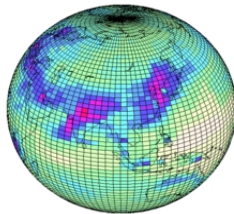
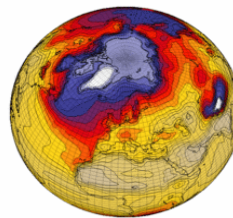
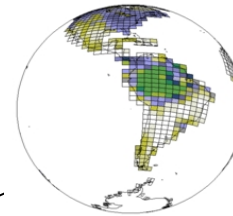
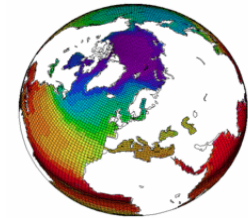
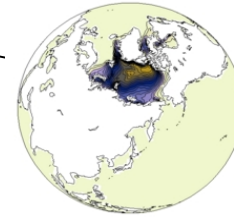




**Institut
Pierre
Simon
Laplace**

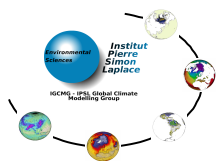
**IGCMG - IPSL Global Climate
Modelling Group**



17 octobre 2011
Groupe de travail Plate-forme

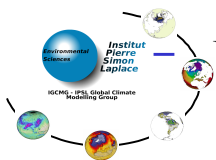
Le modèle couplé de l'IPSL

- Contexte – Présentation
 - IPSL
 - Pôle de modélisation du climat et CMIP5
 - Historique du modèle système Terre IPSL
 - Le modèle IPSLCM5A
 - Les machines utilisées
 - Les autres configurations
 - Les outils
 - La documentation
 - A venir...
- Mode d'emploi
 - Les différentes étapes
 - Les autres outils utiles : monitoring, atlas, nco, cdo,...



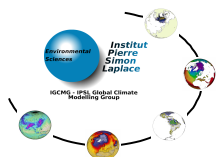
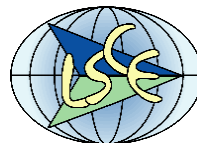
IPSL : Institut PS Laplace

- Fédération de 6 laboratoires, liée à 3 Observatoires des sciences de l'Univers :
 - Laboratoire Atmosphères, Milieux, Observations Spatiales (LATMOS) ,
 - Laboratoire Interuniversitaire des Systèmes Atmosphériques (LISA) ,
 - Laboratoire de Météorologie Dynamique (LMD) ,
 - Laboratoire d'Océanographie et du Climat : Expérimentation et Approches Numériques (LOCEAN),
 - Laboratoire de Physique Moléculaire pour l'Atmosphère et l'Astrophysique (LPMAA),
 - Laboratoire des Sciences du Climat et de l'Environnement (LSCE)
 - 10 tutelles
 - Centre National de la Recherche Scientifique (CNRS : INSU et INEE),
 - Université Pierre et Marie Curie - Paris 6
 - Université Versailles Saint-Quentin en Yvelines
 - Commissariat à l'Energie Atomique (CEA)
 - Institut de Recherche et Développement (IRD)
 - Ecole Normale Supérieure
 - Ecole Polytechnique
 - Centre National d'Etudes Spatiales (CNES).
 - Université Denis Diderot – Paris 7
 - Université Paris 12 Val de Marne
- Directeur : Hervé Le Treut
 - 1000 personnes
 - Structures fédératives/Pôles scientifiques
 - Pôle de modélisation du climat - 80 personnes – Jean-Louis Dufresne





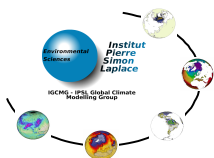
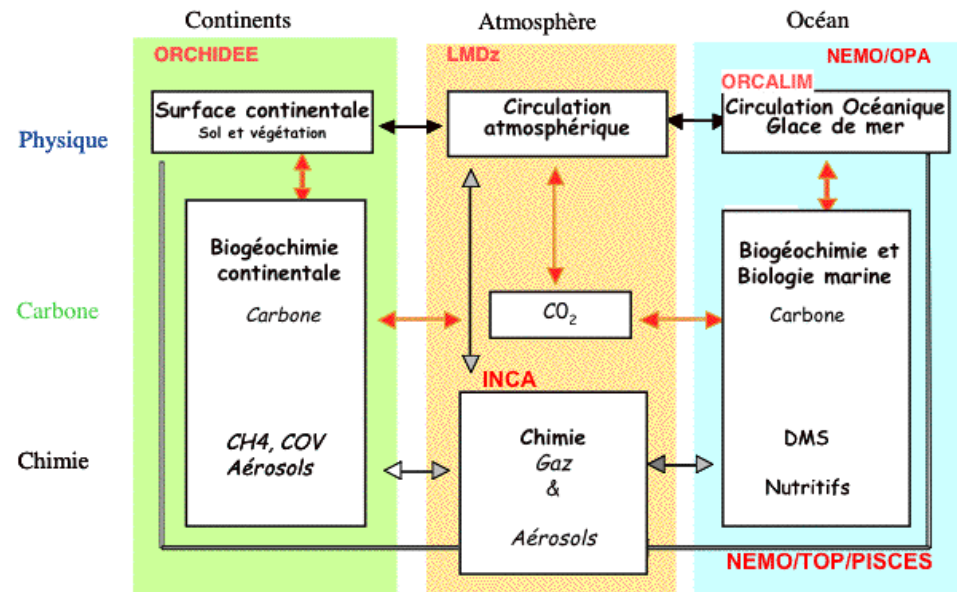
Les laboratoires et les tutelles



Le pôle de modélisation du climat

- Missions :
 - Fédérer les études multidisciplinaires (scientifiques ou techniques) faisant intervenir les composantes du modèle de l'IPSL
 - Identifier et coordonner les simulations de référence
 - Fédérer et rationaliser les moyens, les développements techniques
 - Animation scientifique
- Modèle climat :
 - Atmosphère
 - Océan et glace de mer
 - Surfaces continentales
 - Cycle du carbone
 - Chimie
- IPSLCM5A
- Earth System Model

Le modèle climat IPSL



Pôle de modélisation du climat de l'IPSL : Groupes de travail

Plate-forme de modélisation
(IPSL-ESM)

Physique et dynamique de l'atmosphère et
de la surface
(LMDZ, ORCHIDEE_hydro)

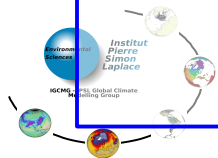
Physique et dynamique de l'océan et de la
glace de mer
(NEMO, LIM)

Interactions atmosphère-océan
(IPSL-CM, différentes résolutions)

Chimie atmosphérique et aérosols
(INCA, INCA_aer, Reprobis)

Cycles biogéochimiques
(PISCES, ORCHIDEE_veget)

Distribution des données



Simulations centennales
(20-21^e siècle)

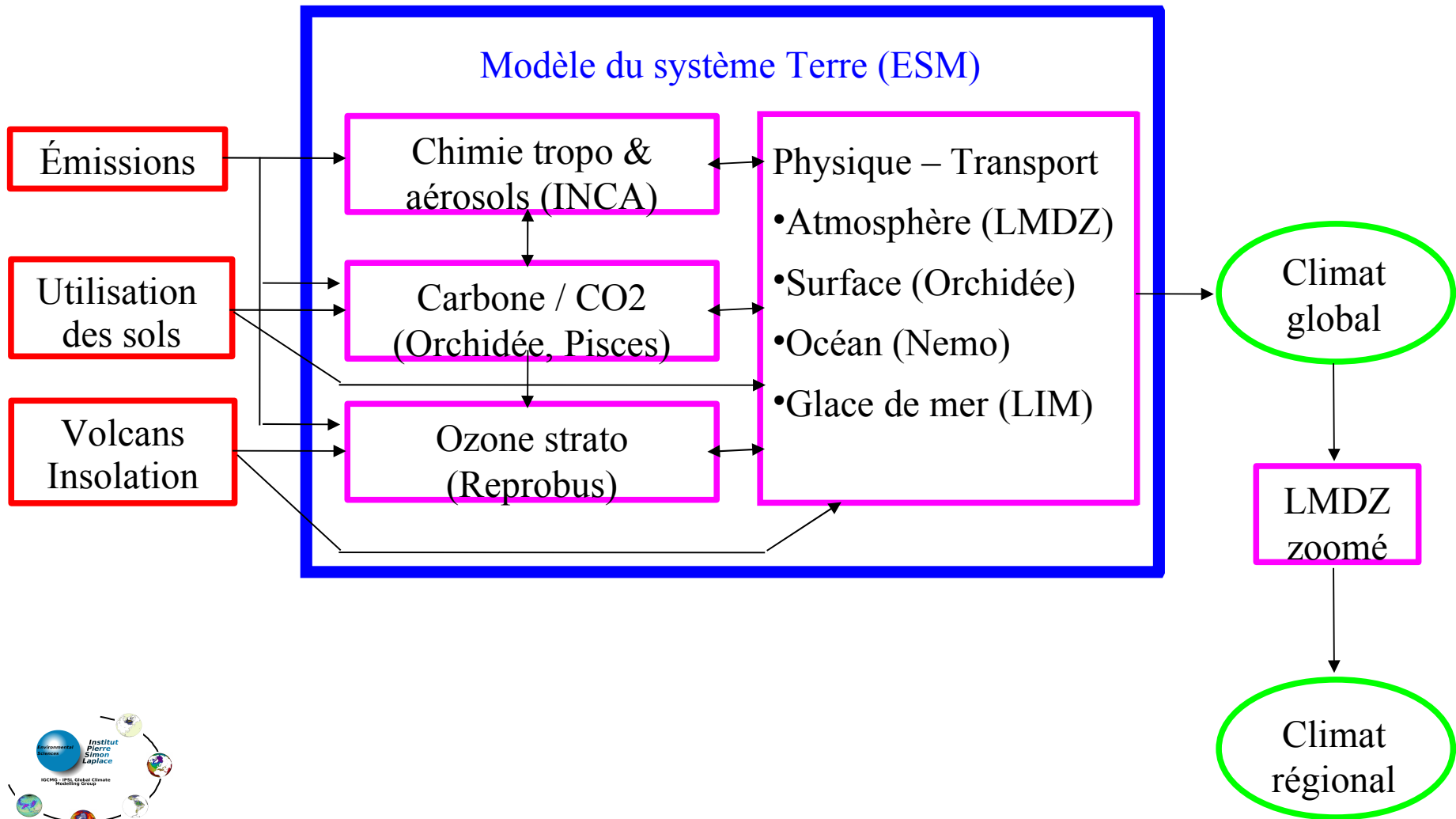
Simulations paléo, dernier millénaire

Simulations saisonnières à décennales

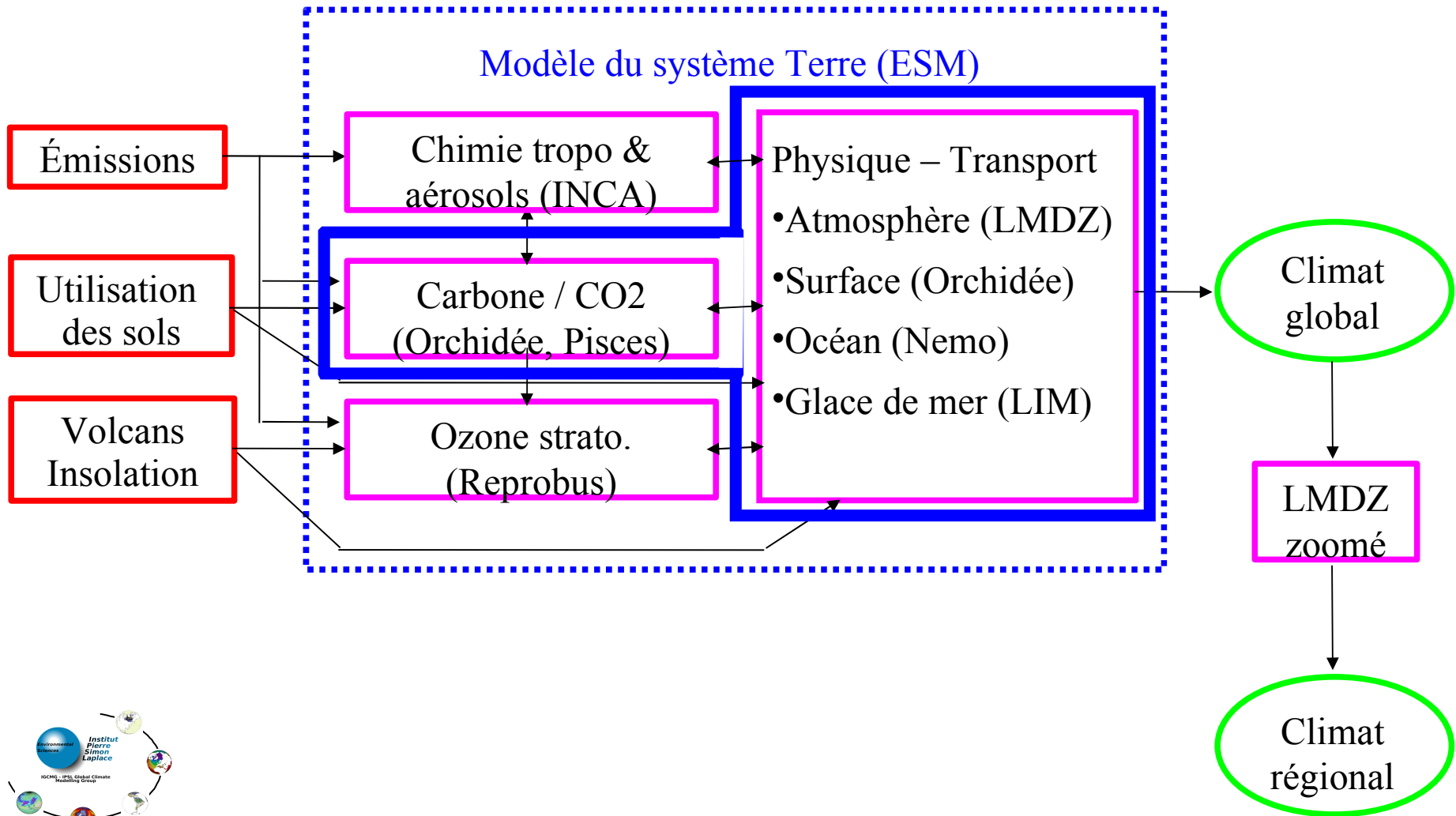
Simulations régionales

Evaluation des modèles
Analyse du climat présent et des
changements climatiques

Modèles de l'IPSL pour CMIP5



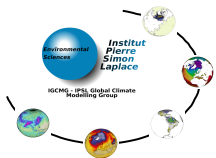
Modèles de l'IPSL pour CMIP5



Le modèle climat de l'IPSL

Définition : plateforme qui permet, sur les *centres de calcul usuels* :

- de *recupérer* des configurations de référence
- de *compiler* :
 - les sources des différentes composantes
 - les interfaces de couplage (océan-atmosphère) et le coupleur
- de *réaliser* une expérience type fournie (y compris fichiers entrée),
- de *suivre* son exécution,
- de *produire* et *stocker* des résultats bruts,
- de *produire, stocker* et *rendre accessible* des ATLAS et analyses systématiques

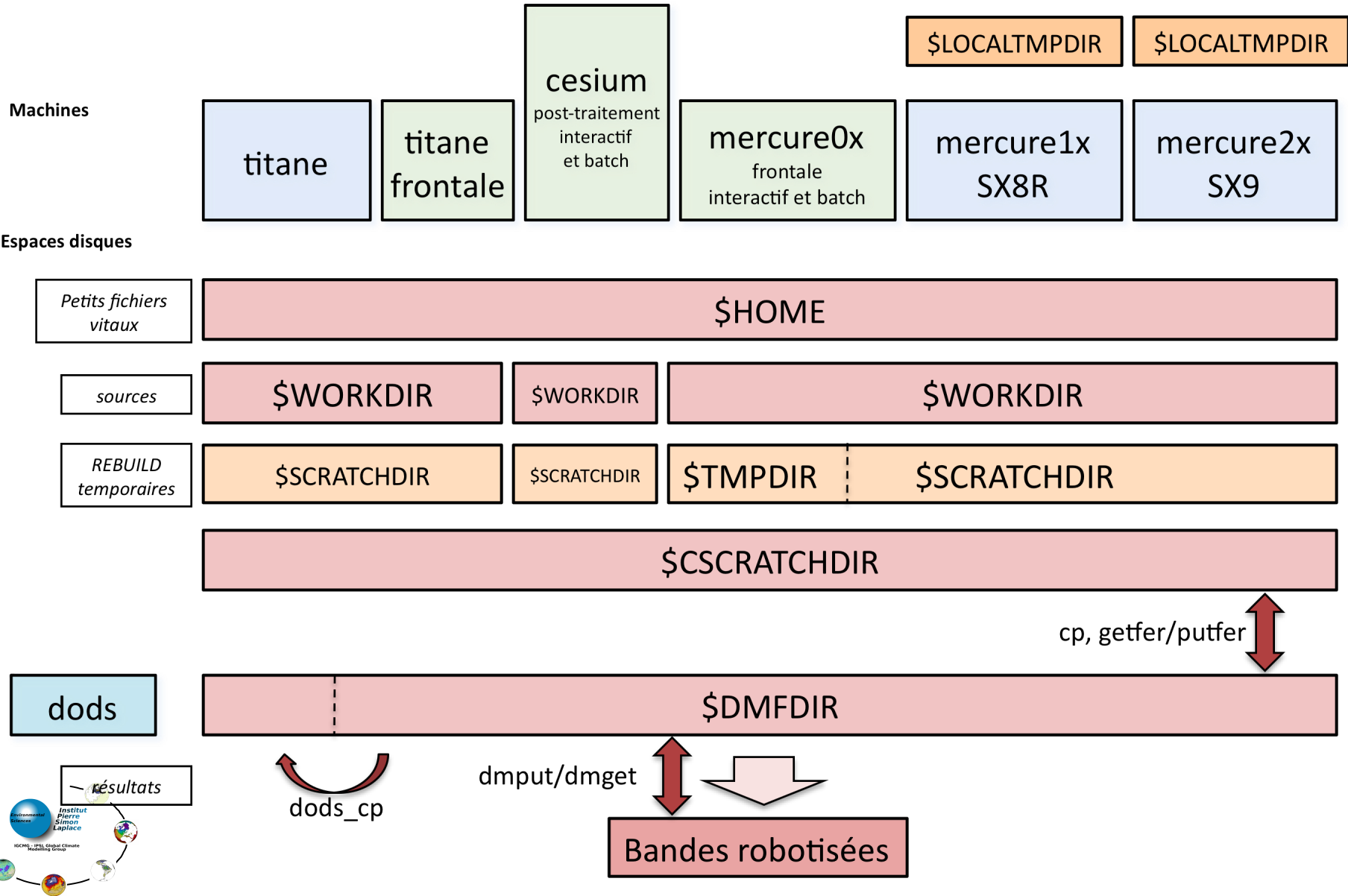


Deux centres de calcul privilégiés

	IDRIS/CNRS	CCRT/CEA
Calculateurs	<ul style="list-style-type: none"> • vargas.idris.fr IBM Power 6, 3584 cœurs • brodie.idris.fr NEC SX-8, été 2006 10 noeuds 80 processeurs 	<ul style="list-style-type: none"> • titane.ccc.cea.fr Bull, 8544 coeurs • mercure.ccc.cea.fr NEC SX-8R, nov 2006 8 noeuds, 64 procs NEC SX-9, avril 2009, 3 noeuds, 48 procs, dédié GIEC 64 processeurs
Connexion	brodie/vargas/ulam.idris.fr (filtrage par adresse)	mercure/titane/cesium.ccc.cea.fr (filtrage par adresse + port 22)
Sources (conseil)	\$WORKDIR	\$WORKDIR/\$SCRATCHDIR
Fichiers	<i>gaya</i>	\$DMFDIR
Post-traitement	ulam	mercure (01), cesium
Serveur DODS	dods.idris.fr	dods.extra.cea.fr/data/
Assistance-Support	assist@idris.fr	hotline.ccrt@cea.fr
	01 69 35 85 55	01 69 26 66 66
	www.idris.fr	www-ccrt.cea.fr + Intranet (firefox sur titane)



Les machines et espaces disques du CCRT en une figure



A savoir sur le CCRT

<http://forge.ipsl.jussieu.fr/igcmg/wiki/ModipslBeginner1>

PATH=\$PATH:/home/cont003/p86ipsl/fcm/bin

pour accéder à fcm. Vérifier par `which svn` et `which fcm`

Remplir le fichier `~/forward` avec son adresse mail pour recevoir les messages de fin de simulation.

Pour avoir accès aux différents outils, passer par les modules. Exemple:

`module load netcdf`

`module load ferret`

`module load cdo`

Pour avoir un environnement complet nous vous recommandons de linker vers les fichiers

`~p86ipsl/.atlas_env_nomachine_ksh` (ou `_csh`)

Ex : `~p86ipsl/.atlas_env_titane_ksh`

On ne travaille pas sur le `$DMFDIR` --> on transfère les fichiers sur le `$SCRATCHDIR` ou `$CSCRATCHDIR` avant de les analyser

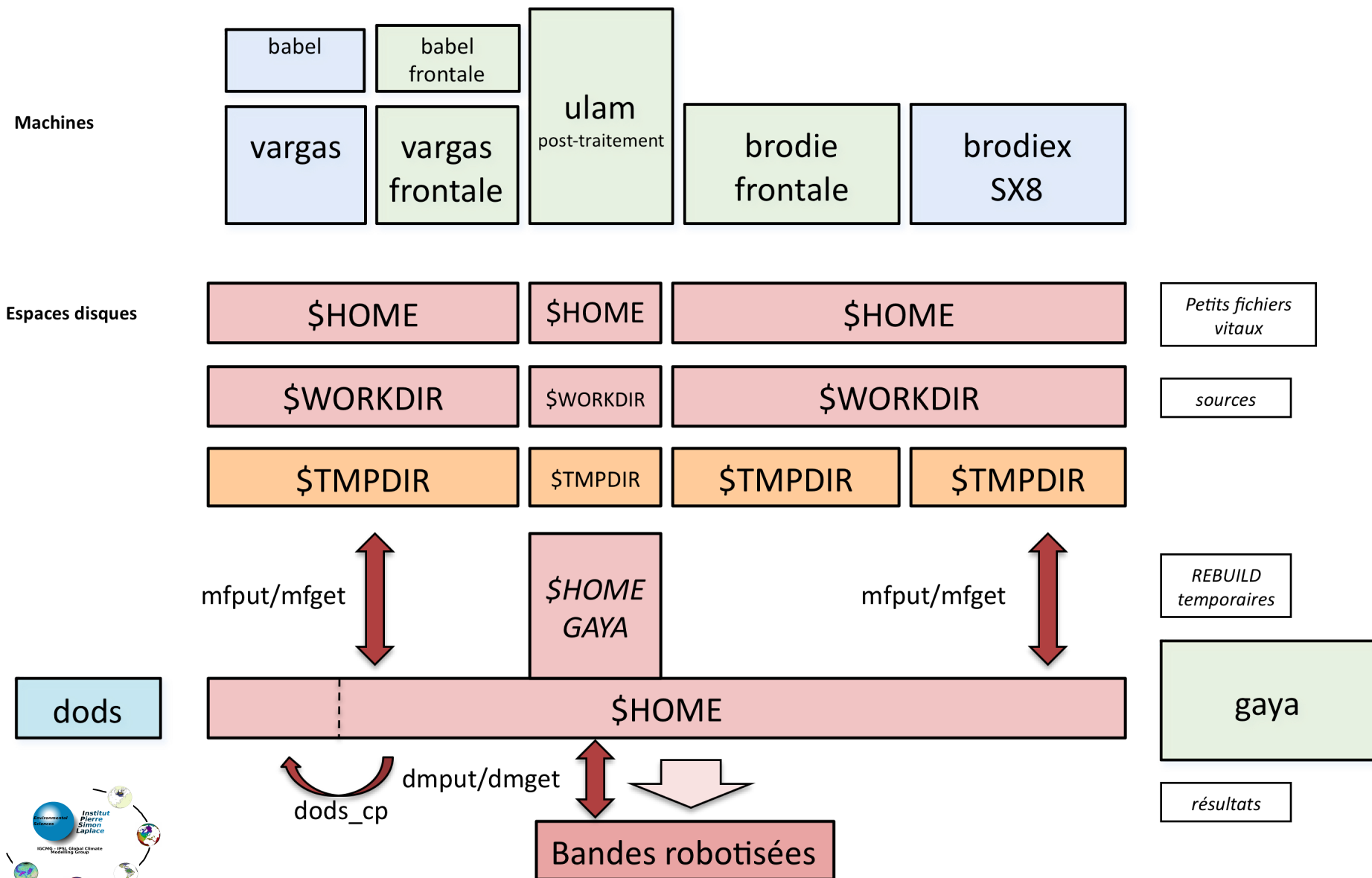
Remarque : SX8 et SX9 sont 2 machines différentes :

SX9 : `module load SX9 ; module load netcdf_sx9 ;`

SX8 : `module load SX8 ; module load netcdf_sx8 ;`

(`module unload ...`)

Les machines et espaces disques de l'IDRIS en une figure



A savoir sur l'IDRIS

<https://forge.ipsl.jussieu.fr/igcmg/wiki/ModipslBeginner1>

PATH=/Txlocal/pub/svn/svn-1.3.1/bin:/home/rech/psl/rpsl035/fcm/bin:\$PATH

pour accéder à svn et à fcm. Vérifier par `which svn` et `which fcm`

\$WORKDIR sur *brodie* peut être **étendu largement** (50 Go pour le groupe par exemple). Le demander à l'assistance. Pour vérifier l'occupation et la taille : `quota_u -w`

sur *brodie*, remplir le fichier *.rhosts* avec `ulam`

Faire marcher les **transferts brodie --> gaya** par `mfget/mfput`. *Ftuas* sur *ulam* pour faire connaître le mot de passe gaya à brodie et à toutes les machines.

Pour les **accès dods**, il faut lancer une commande `mfdods` sur *gaya*. Cela crée le répertoire, 24h après au plus tard. Accès web : <http://dods.idris.fr/login>

Pour les **accès dods**, sur *ulam* la commande `rsh gaya pwd` doit marcher

Pour donner les accès à tous (755 ou `drwxr-xr-x`) au **WORKDIR** de brodie, il faut **demandeur à l'assistance** IDRIS pour le niveau `/u/rech/grp`. Idem pour `/home_b/rech/grp` sur rhodes.

Sur *ulam* il faut explicitement charger les outils graphiques :

```
module load netcdf
```

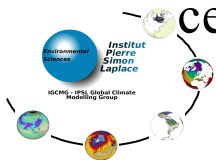
```
module load cdo
```



Les configurations cohérentes avec IPSLCM5A

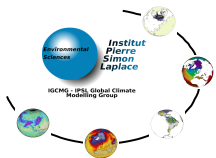
- Les forcés :
 - LMDZOR_v4 : J. Ghattas
 - ORCHIDEE_OL : M. Mancip
 - LMDZORINCA : A. Cozic
 - ORCA2_LIM_PISCES : C. Ethé
 - IPSLCM5A_C : P. Cadule
- Les déclinaisons du Earth System model (à venir)
 - LMDZ_REPROBUS : J. Ghattas
 - IPSL_ESM_INCA : A. Cozic
 - IPSL_ESM_REPROBUS : M. Marchand

Recommandation : prévenir lors de nouvelles études basées sur une de ces configurations



A venir...

- Futur ultra proche, déjà utilisé :
 - parallélisation mixte MPI OpenMP LMDZ-ORCHIDEE => machines cibles scalaires SMP type titane ou vargas avec plus d'efficacité
 - Grand challenge au CINES, SGI, > 2000 procs
 - Couplé LMDZ 1/3°- OASIS -NEMO 1/4°
 - Nouvelle physique LMDZ : IPSLCM5B
- Futur moins proche
 - Serveur IO dans les modèles IPSL
 - Coupleur OASIS4 : plus de parallélisme
 - Utilisation machines scalaires MPP ~1000 procs



Environnement

Récupération de la configuration

Serveurs CVS/SVN

Visualisation/comparaison des résultats

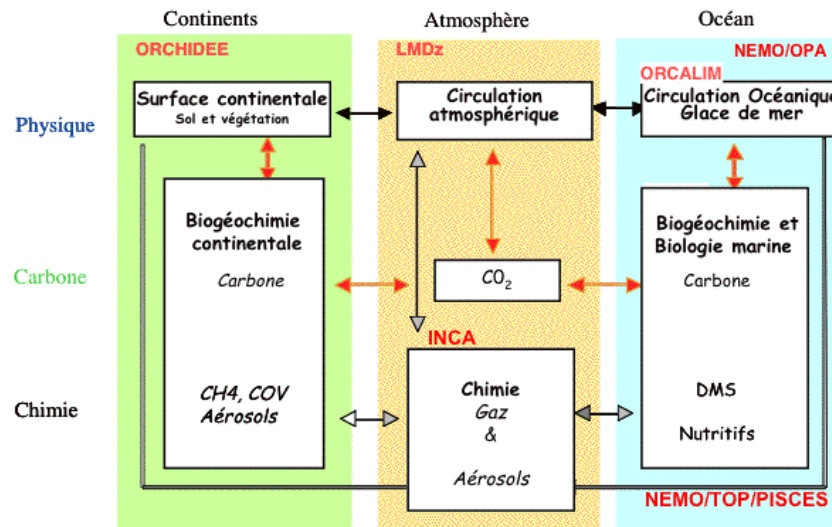
Assemblage du modèle

IOserver

Accès aux résultats

LibIGCM

Formation



Modipsl

Compilation

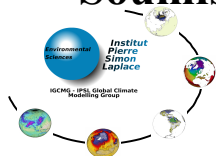
Support

Machines

Soumission/Exécution

Description d'une expérience

Documentation



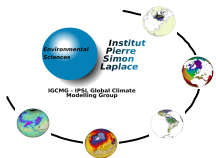
Les outils : MODIPSL, IOIPSL, Rebuild et IOserver

- **Modipsl** : outil d'extraction, de préparation en fonction de la machine, de compilation des modèles pour créer les exécutable.

Introduction – FAQ :

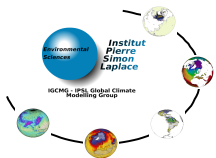
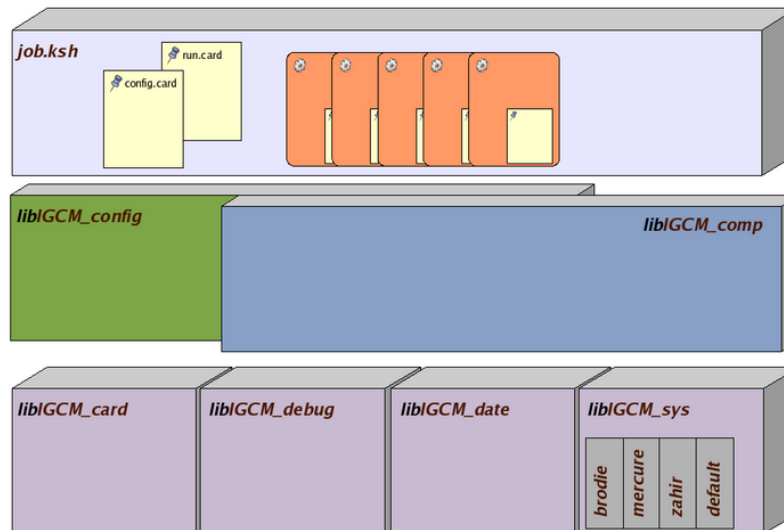
<http://forge.ipsl.jussieu.fr/igcmg/wiki/ModipslBeginner1>

- **IOIPSL** : librairie commune qui gère les Entrées-Sorties (fichiers de sorties, restarts) au format NetCDF dans les modèles de l'IPSL.
- **Rebuild** : outil pour recombinaison des fichiers créés par sous-domaine, tourne maintenant sur les frontales, en asynchrone
- **IOserver** : outil gérant les Entrées/Sorties, piloté par fichier xml, capable de tourner en mode asynchrone sur 1 ou plusieurs procs.



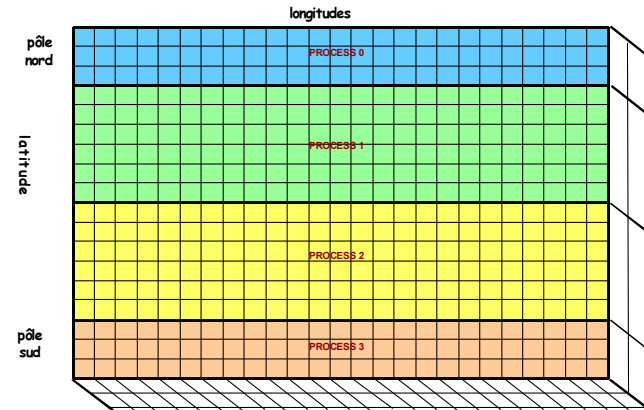
Les outils : les scripts libIGCM

- libIGCM : ensemble de scripts de lancement de simulation et de post-traitement **modulaires** et **portables**
- Inclut le job maitre et de nombreux jobs auxiliaires de post-traitements
- Documentation : <http://forge.ipsl.jussieu.fr/libigcm>



Parallélisme

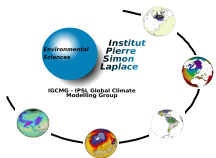
- Utilisation de plusieurs processeurs pour réduire le temps d'exécution
- Parallélisation MPI pour LMDZ/Orchidee/INCA
- Parallélisation MPI possible pour NEMO
- Machines cibles :
 - Vectoriel et parallélisme modéré $\alpha(10)$, MPI : Mercure (NEC SX-8R et SX-9), Brodie (NEC SX-8)
 - Scalaire SMP parallélisme massif $\alpha(100)$: Platine/Titane (BULL), vargas (IBM), performances améliorées mixte MPI/OpenMP
- Performances : 1 an couplé en ORCA2xLMD96x95x39



Machine	Brodie	Vargas	SX9	titane
nb procs	4 (3+1+1)	32 (26+5+1)	4 (3+1+1)	32 (26+5+1)
Temps réel	4h	4h	3h	4h
Temps CPU	17h	130h	11h	128h

Documentation générale

- Wiki Pôle : <http://forge.ipsl.jussieu.fr/igcmg>
 - Introduction générale IGCMG, accessible à tous
 - Accès outils multiples
 - Accès sources SVN
 - Accès aux tickets d'incident
 - Accès aux pages wiki
 - Contrainte : être inscrit dans le projet (demande aux administrateurs) pour pouvoir **modifier** wiki, tickets et sources.
 - Machine commune de gestion des projets - Olivier Thauvin (LATMOS)



Documentation

forge.ipsl.jussieu.fr/igcmg



WIKI

Sources SVN

Navigation bar with search box and menu items:

- Search
- Login
- Help/Guide
- About Trac
- Preferences
- Wiki
- Timeline
- Roadmap
- Browse Source
- View Tickets
- Search
- Start Page
- Index
- History
- Last Change

Welcome to IGCMG Wiki page

- ⇒ [Official site for IGCMG](#) on ⇒ [IPSL site](#)
- Official site for IGCMG with AR4 model documentation : <http://igcmg.ipsl.jussieu.fr/>
- Intranet site (private access) for IGCMG : ⇒ http://wiki.ipsl.jussieu.fr/wiki_ipsl/PoleModelisationClimat

News

- [News](#) : all news of IGCMG group

CMIP5 at IPSL

- [CMIP5 at IPSL](#)

Documentation

- [Documentation](#) : all documentation about IPSL Earth System model

The CPLIPSL group

- Direct access to [CplIpsl](#) page to have more information about the status of the IPSL global climate model.

Projects



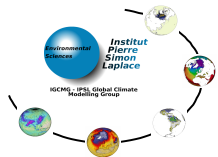
Tickets

Welcome to IGCMG Wiki page

- [News](#)
- [CMIP5 at IPSL](#)
- [Documentation](#)
- [The CPLIPSL group](#)
- [Projects](#)
- [IGCMG projects](#)
- [French research agency \(ANR\) projects](#)
- [Europeans projects](#)
- [Animations](#)

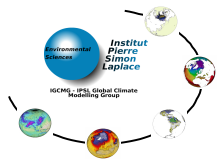
2ème partie

Utilisation et démonstration



Connexion

Frontale



Gestion des sources des composantes

Serveur cvs/svn

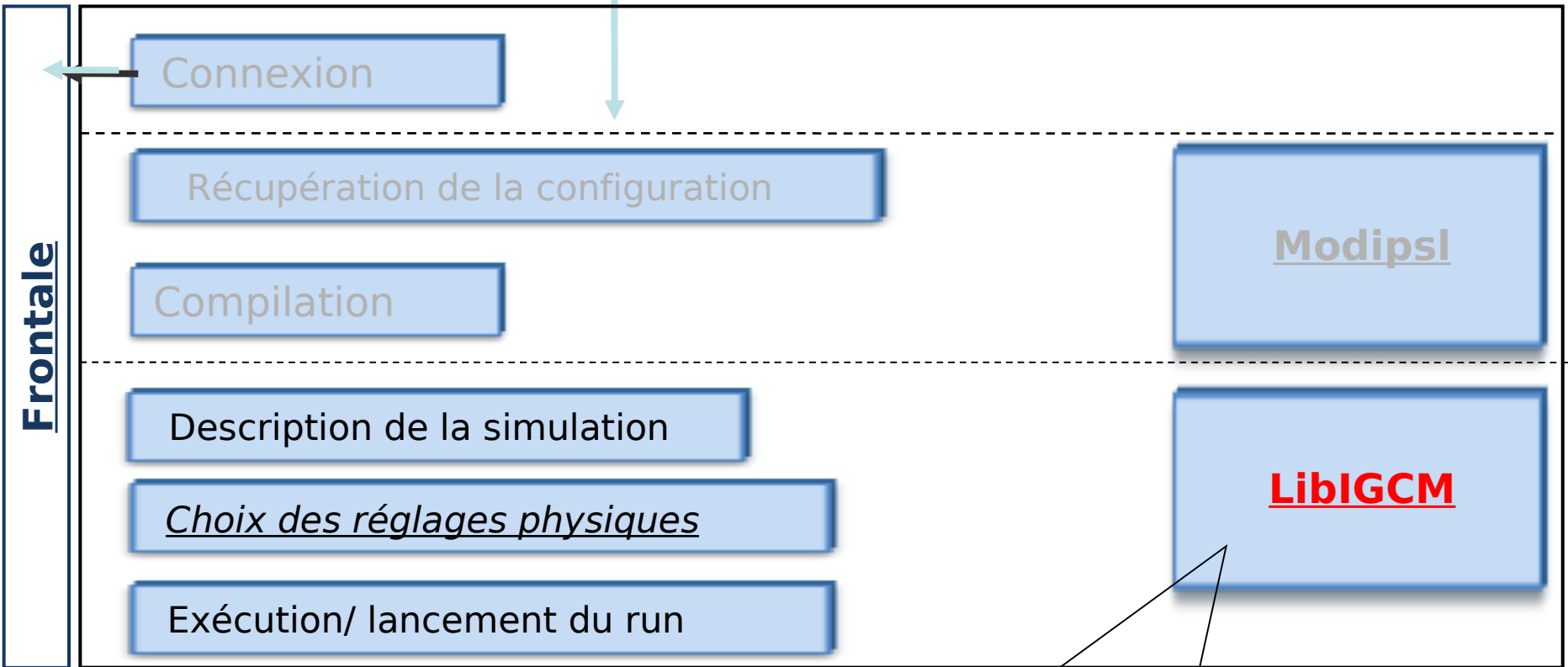
Connexion

Récupération de la configuration

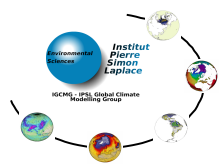
Compilation

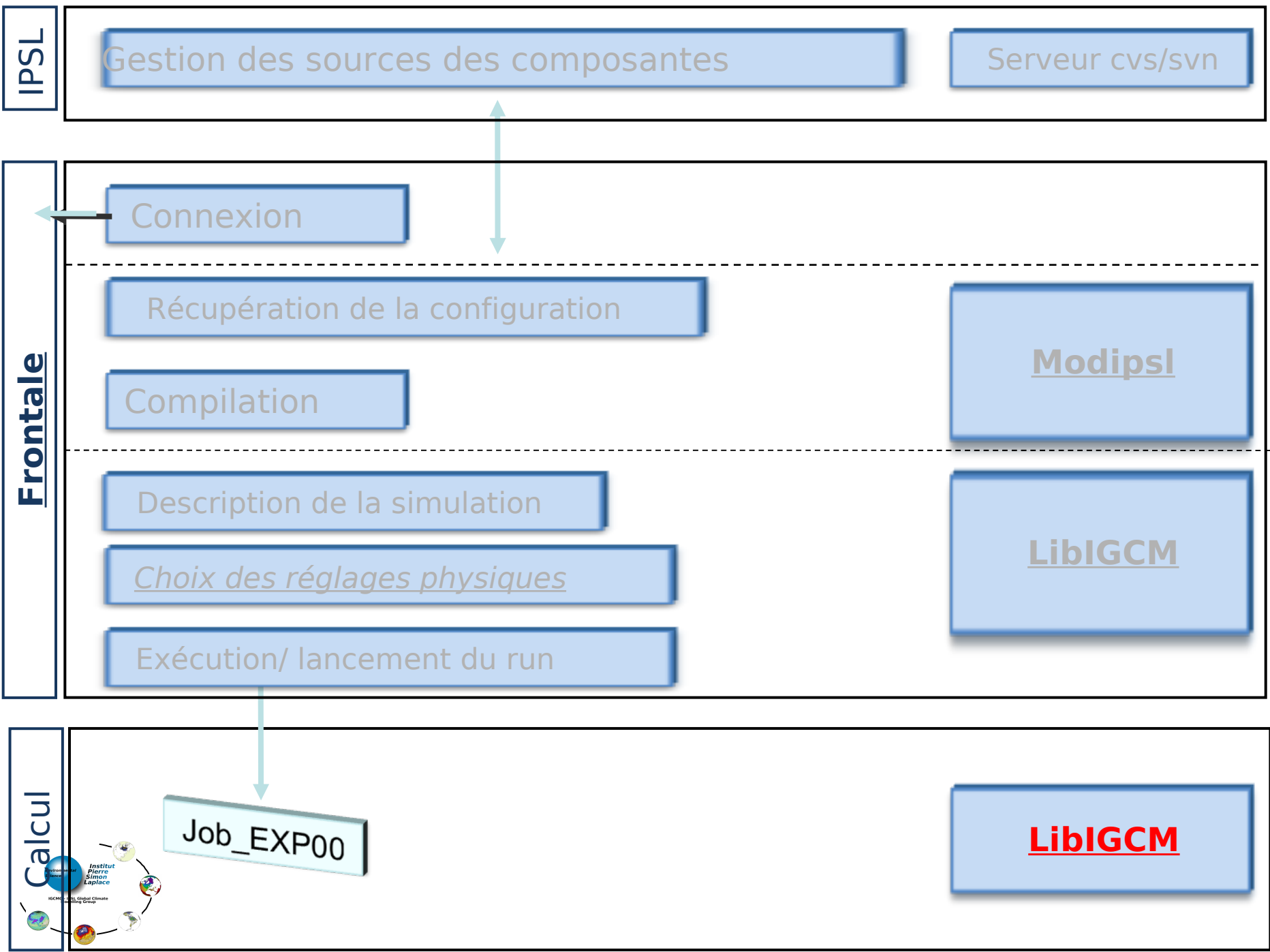
Modipsl

Ensemble de scripts qui va permettre de faire ces étapes sur un ensemble de machines.



ensemble de scripts de lancement de simulation et de post-traitement **modulaires et portables**





IPSL

Gestion des sources des composantes

Serveur cvs/svn

Frontale

Connexion

Récupération de la configuration

Compilation

Modipsl

Description de la simulation

Choix des réglages physiques

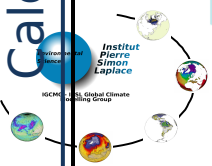
Exécution/ lancement du run

LibIGCM

Calcul

Job_EXP00

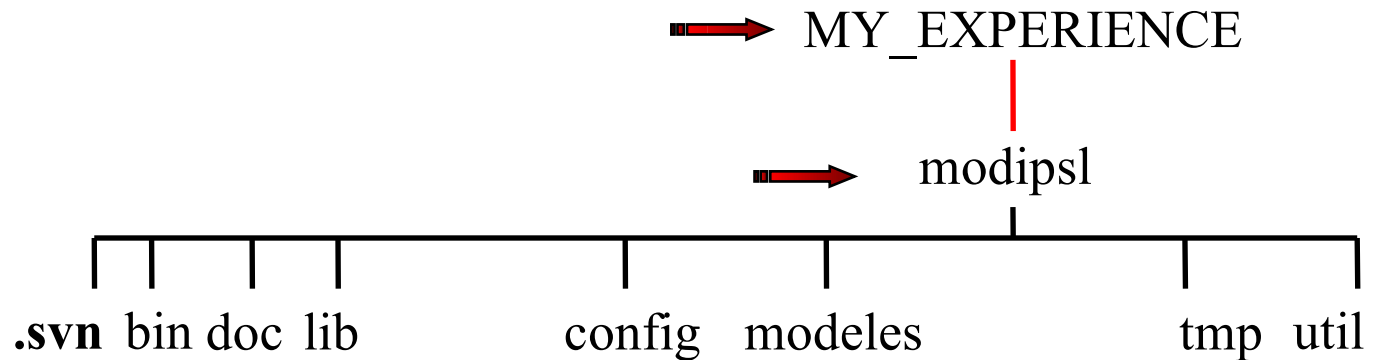
LibIGCM



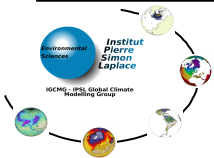
Récupérer une configuration

Accès à *modipsl* depuis le cert ou l'idris (SVN)

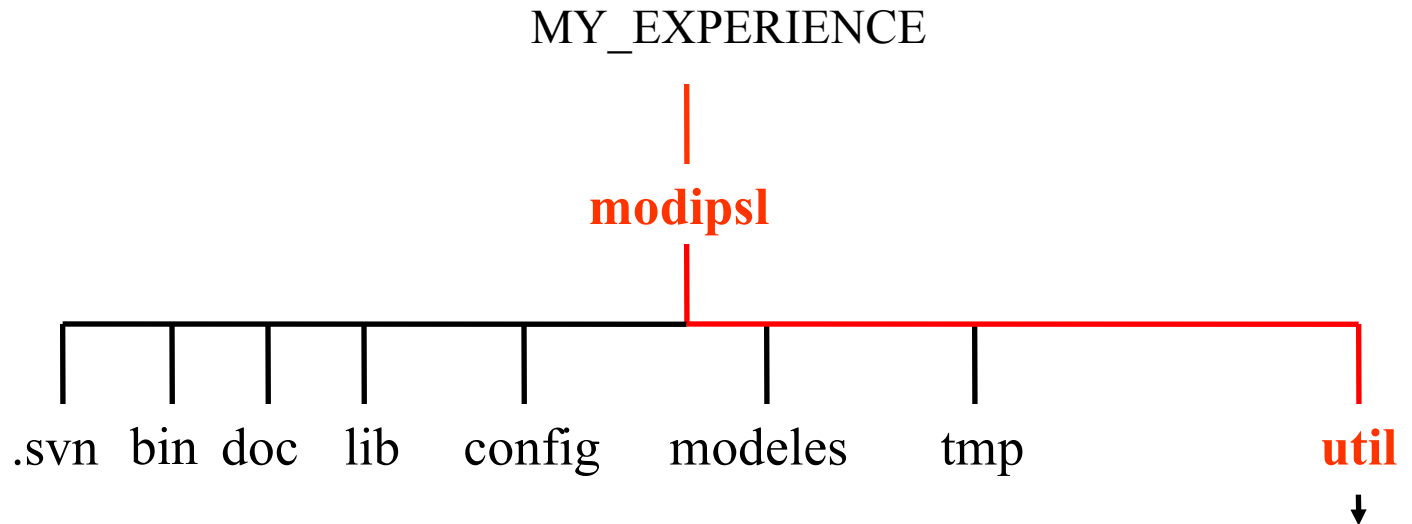
```
mkdir MY_EXPERIENCE
cd MY_EXPERIENCE
svn co http://forge.ipsl.jussieu.fr/igcmg/svn/modipsl/trunk modipsl
```



```
cd modipsl/util/
```



Récupérer une configuration



Définition pour chaque configuration de leurs
composantes et de leurs tags associés



• *mod.def*

Extraction des modèles validés disponibles



• **model**

Installation et configuration des Makefiles

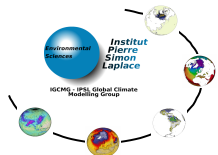


• **ins_make**

Installation et configuration des scripts de lancement



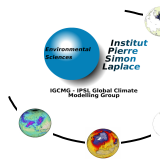
• **ins_job**



Récupérer une configuration

- fichier mod.def -

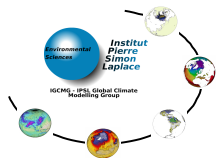
```
##- $Id$
##-----
##-This file is the definition file of the script "model".
##-----
##- Each model is defined by
##- (prefix #-H-) model informations,
##- (prefix #-M-) the email address of the model manager,
##- (prefix #-C-) elements to extract for the model, in the order :
##-   name of the component in the repository
##-   tag/revision of the component
##-   index of the repository in the server table
##-   installation path in the local working directory
##-   local working directory in modipsl
##- (prefix #-S-) containing the control system and server address.
##-
##- The tag "?" correspond to the default model version.
##- Invoking "model" with -H overrides any tag with "?".
##-----
##- Repository informations
##-
##-S- 1 cvs anonymous@cvs.ipsl.jussieu.fr:/home/ioipsl/CVSROOT
##-S- 2 cvs sechiba@cvs.ipsl.jussieu.fr:/home/ssipsl/CVSREP
##-S- 3 cvs lmdzbrowse@cvs.lmd.jussieu.fr:/home/cvsroot
##-S- 4 cvs opa@cvs.ipsl.jussieu.fr:/home/opalod/CVSROOT
##-S- 5 cvs nemo@cvs.ipsl.jussieu.fr:/home/opalod/NEMOCVSROOT
##-S- 6 cvs inca@cvs.ipsl.jussieu.fr:/home/incaipsl/CVSROOT
##-S- 7 svn http://forge.ipsl.jussieu.fr/nemo/svn
##-S- 8 svn http://forge.ipsl.jussieu.fr/igcmg/svn
##-S- 9 svn --username inca http://forge.ipsl.jussieu.fr/inca/svn
##-S- 10 svn http://forge.ipsl.jussieu.fr/libigcm/svn
##-S- 11 svn http://svn.lmd.jussieu.fr/LMDZ
##-S- 12 svn http://forge.ipsl.jussieu.fr/ioserver/svn
##-S- 13 svn http://forge.ipsl.jussieu.fr/fcm/svn
```



Récupérer une configuration

- fichier mod.def -

```
#-H- IPSLCM5A IPSLCM5A coupled configuration
#-H- IPSLCM5A CMIP5 version 30/04/2010
#-H- IPSLCM5A NEMO svn branches/CMIP5_IPSL 1854
#-H- IPSLCM5A XMLF90 svn trunk revision 54
#-H- IPSLCM5A XMLIO_SERVER svn trunk revision 54
#-H- IPSLCM5A IOIPSL/src svn tags/v2_2_0
#-H- IPSLCM5A LMDZ4 trunk revision 1374
#-H- IPSLCM5A ORCHIDEE tag orchidee_1_9_4_2
#-H- IPSLCM5A OASIS3 tag ipslcm5a
#-H- IPSLCM5A IPSLCM5A svn
#-H- IPSLCM5A libIGCM trunk revision 265
#-M- IPSLCM5A arnaud.caubel@lsce.ipsl.fr
#-C- IPSLCM5A IOIPSL/tags/v2_2_0/src          HEAD           8 IOIPSL/src modeles
#-C- IPSLCM5A ORCHIDEE                      orchidee_1_9_4_2  2 .             modeles
#-C- IPSLCM5A OASIS3                        ipslcm5a          1 prism        .
#-C- IPSLCM5A LMDZ4/trunk                   1374             11 LMDZ4         modeles
#-C- IPSLCM5A CONFIG/IPSLCM/IPSLCM5A      HEAD             8 IPSLCM5A     config
#-C- IPSLCM5A trunk/libIGCM                265              10 libIGCM      .
#-C- IPSLCM5A branches/CMIP5_IPSL/NEMO    1854             7 .            modeles
#-C- IPSLCM5A branches/CMIP5_IPSL/UTIL    1854             7 .            modeles
#-C- IPSLCM5A XMLF90                       54               12 .            modeles
#-C- IPSLCM5A XMLIO_SERVER/trunk          54               12 XMLIO_SERVER modeles
```



Récupérer une configuration

commande model

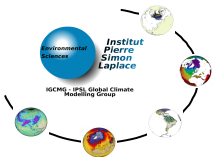
- 1) `./model -h` --> liste des configurations de modèles disponibles
- 2) `./model -h config` --> pour une configuration donnée donne la liste des modèles la composant
- 3) `./model config` --> extraire une configuration donnée

Extraire IPSLCM5A :

```
cd modipsl/util  
./model IPSLCM5A
```

Attention:

3 mots de passe --> celui d'Orchidee, anonymous puis celui de NEMO.
Pour NEMO s'enregistrer <http://www.nemo-ocean.eu/user/register>



Récupérer une configuration

MY_EXPERIENCE

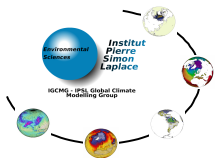
modipsl

bin .svn doc lib **libIGCM** **prism** **config** modeles tmp util



IPSLCM5A

IOIPSL UTIL ORCHIDEE NEMO LMDZ4



Récupérer une configuration

Installation des makefiles

```
cd modipsl/util  
./ins_make
```

MY_EXPERIENCE

modipsl



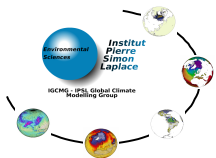
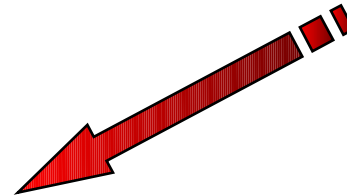
• AA_make.gdef

IPSLCM5A



- AA_make.ldef
- AA_make

Makefile



Récupérer une configuration compilation

Lors de la compilation il faut choisir la résolution à laquelle le modèle tournera

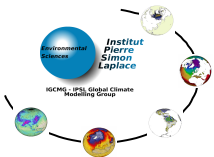
Pour connaître les différents choix il faut ouvrir le Makefile

```
ORCA2xLMD9695 : libioips1 oasis3 liborchidee orca2 lmdz96x95x19 verif
echo "ORCA2xLMD9695" >.resol
echo "RESOL_ATM_3D=96x95x19" >>.resol
echo "$(LIB_MPI)" >.libmpi
```

```
ORCA2xLMD9695-L39 : libioips1 oasis3 liborchidee orca2 lmdz96x95x39 verif
echo "ORCA2xLMD9695-L39" >.resol
echo "RESOL_ATM_3D=96x95x39" >>.resol
echo "$(LIB_MPI)" >.libmpi
```

Remarque : pour chaque configuration il existe une résolution par défaut.

```
if [ -s ./resol ] ; then $(M_K) `head -1 .resol` ; else $(M_K)
ORCA2xLMD9695-L39 ; fi
```



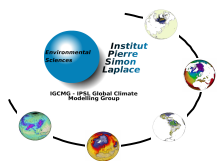
Récupérer une configuration compilation

```
cd modips1/config/IPSLCM5A  
gmake ORCA2xLMD9695-L39
```

Création d'un fichier *.resol* à la fin de la compilation

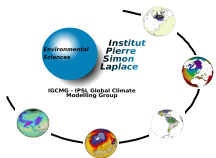
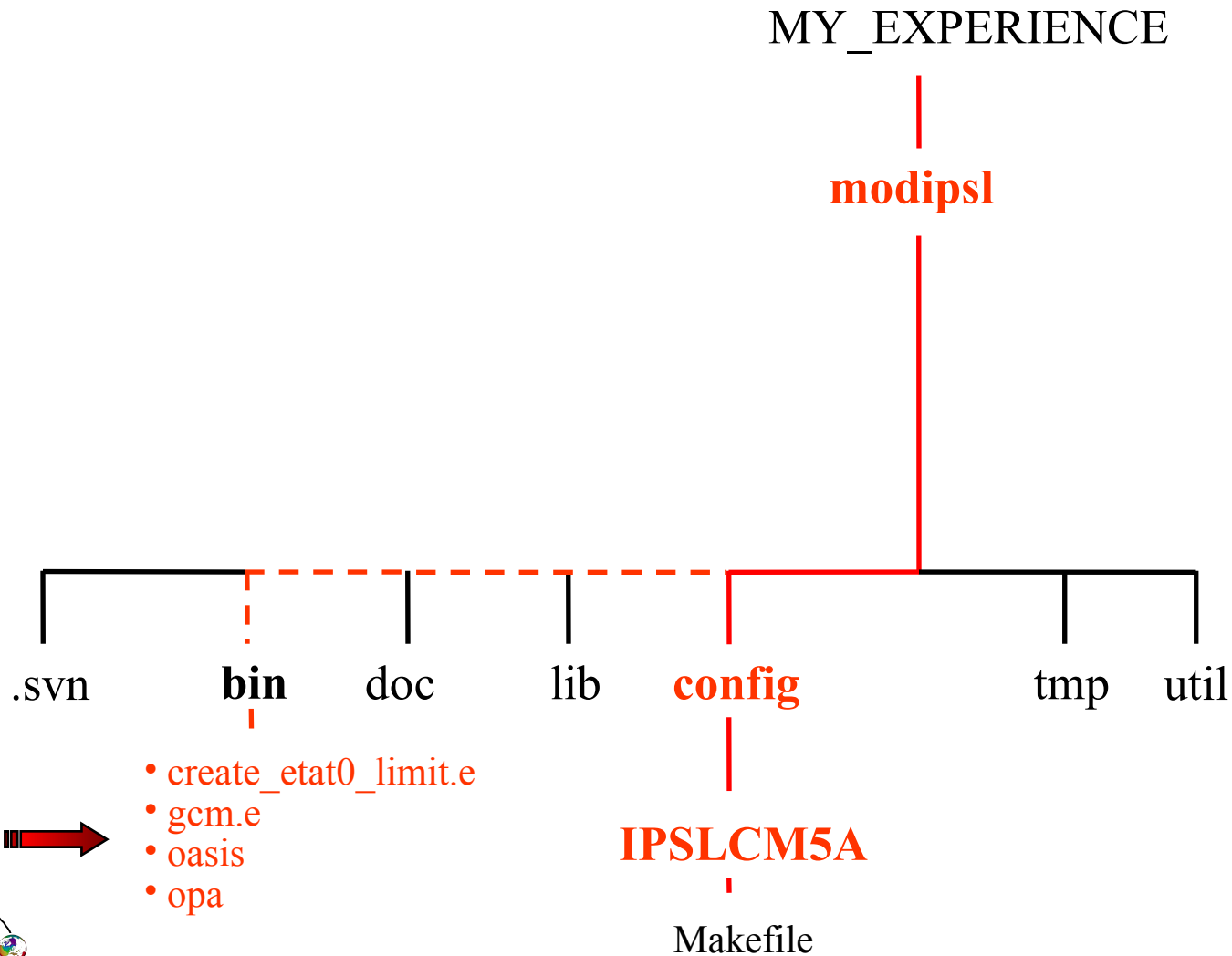
```
cd modips1/config/IPSLCM5A  
vi .resol  
  >> ORCA2xLMD9695-L39  
  >> RESOL_ATM_3D=96x95x39
```

Si la résolution indiquée dans le fichier *.resol* vous convient vous pouvez recompiler avec uniquement la commande « `gmake` »



Attention si vous recompilez alors qu'une simulation est en cours :
lors du prochain chaînage la simulation utilisera le nouvel exécutable

Récupérer une configuration



Récupérer une configuration

Installation de l'expérience type

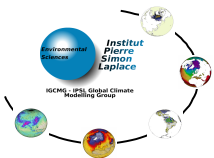
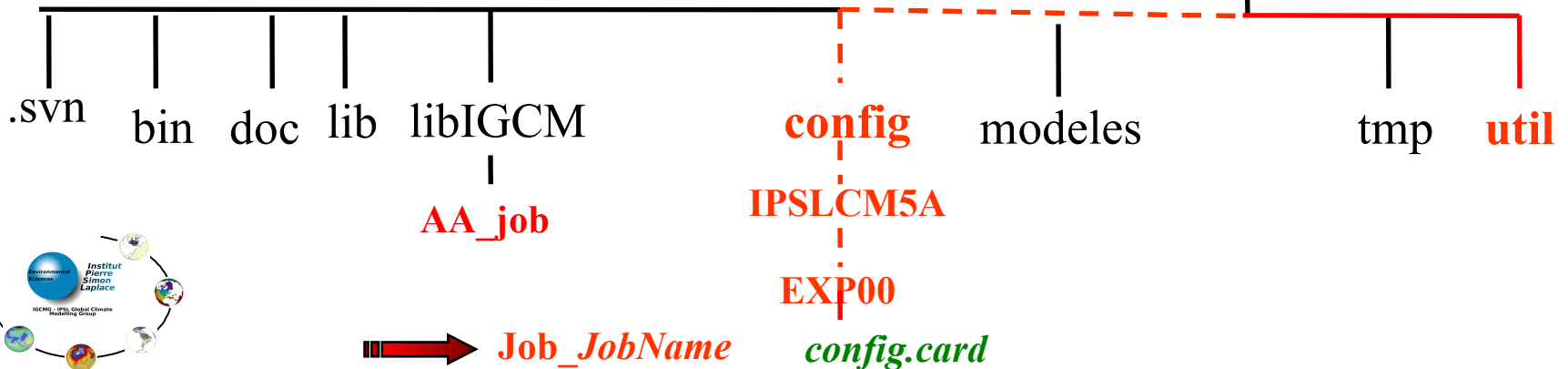
Préparation du config.card et des fichiers COMP/*.card

Création du job

```
cd modipsl/util/  
./ins_job
```

MY_EXPERIENCE

modipsl



Récupérer une configuration

Installation de l'expérience type
Création des jobs de post-traitement

```
cd modips1/util/  
./ins_job
```

MY_EXPERIENCE

modips1

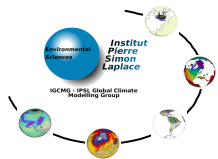
config

modeles

libIGCM

atlas_LMDZ.job
atlas_ORCHIDEE.job
atlas_ORCA_LIM.job
create_ts.job
create_se.job
monitoring.job

AA_atlas_LMDZ
AA_atlas_ORCHIDEE
AA_atlas_ORCA_LIM
AA_create_ts
AA_create_se
AA_monitoring



Récupérer une configuration

Soumission du Job de lancement

```
cd modips1/config/IPSLCM5A/EXP00  
ccc_msub Job_jobname
```

MY_EXPERIENCE

modips1

config

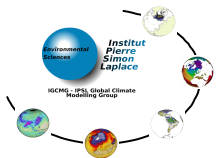
IPSLCM5A

EXP00

Job_JobName **COMP** **PARAM**

lim.card
lim.driver
lmdz.card
lmdz.driver
oasis.card
oasis.driver
opa.card
opa.driver
orchidee.card
orchidee.driver

dynami.param
gcm.def
geogram.param
inice.param
namcouple
namelist
offline.def
orchidee.def
output.param
physiq.def
run.def
run.param.li
thermo.param



Récupérer, compiler et lancer une configuration

1. Accès à MODIPSL

```
svn co http://forge.ipsl.jussieu.fr/igcmg/svn/modipsl/trunk modipsl
```

2. Accès à IPSLCM5A

```
cd modipsl/util ; ./model IPSLCM5A
```

3. Installation des Makefiles

```
cd modipsl/util ; ./ins_make
```

4. Compilation

```
cd modipsl/config/IPSLCM5A ; gmake + resolution choisie
```

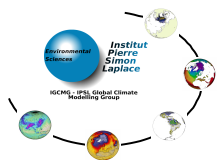
5. Installation de l'expérience type (et post-traitements)

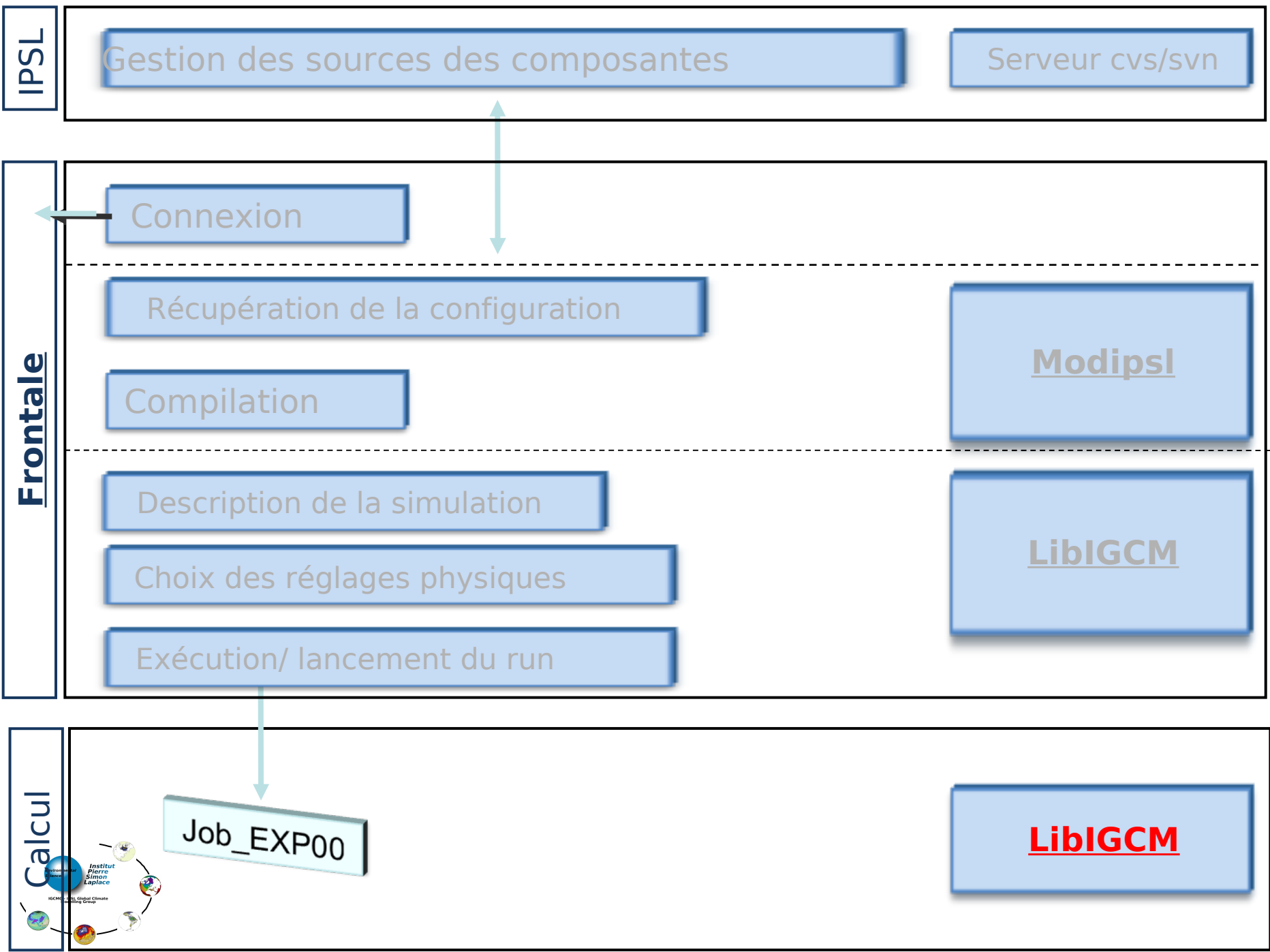
Modifier «*JobName*» dans `../config/IPSLCM5A/EXP00/config.card`

```
cd modipsl/util ; ./ins_job
```

6. Soumission du Job de lancement

```
cd modipsl/config/IPSLCM5A/EXP00; ccc_msub Job_JobName
```





IPSL

Gestion des sources des composantes

Serveur cvs/svn

Frontale

Connexion

Récupération de la configuration

Compilation

Modipsl

Description de la simulation

Choix des réglages physiques

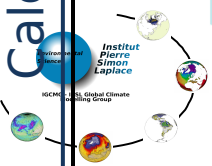
Exécution/ lancement du run

LibIGCM

Calcul

Job_EXP00

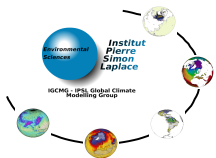
LibIGCM



scripts : libIGCM

Infrastructure commune cohérente de script :

- *Job_Jobname*
 - Un job de soumission de la simulation
- config.card :
 - une fiche descriptive d'une configuration pour une simulation donnée
- COMP :
 - Des couples de fichiers *card* et *driver* **décrivent les fichiers et programment le fonctionnement** de chaque composante d'une configuration
- PARAM :
 - Des fichiers de paramètres **des différentes composantes**
- run.card (run.card.init):
 - Une fiche d'information sur la simulation en cours d'exécution
- libIGCM :
 - libIGCM/libIGCM_card, libIGCM_comp, libIGCM_config, libIGCM_date, libIGCM_debug, libIGCM_post, libIGCM_sys.ksh : des bibliothèques de **fonctions** en ksh utilisées par les jobs
 - libIGCM/libIGCM_sys/libIGCM_sys_vargas.ksh, libIGCM_sys_titane.ksh, ... : des **fonctions** système spécifique à chaque machine



Un peu plus en détail...

Répertoire EXP00 prêt :

- COMP/* : information sur les composantes
- config.card : fichier de configuration de la simulation
- Job_EXP00 : Job à soumettre
- PARAM/* : fichiers de configuration des modèles
- run.card.init : fichier de suivi original

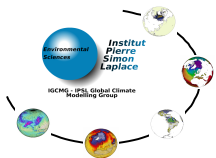
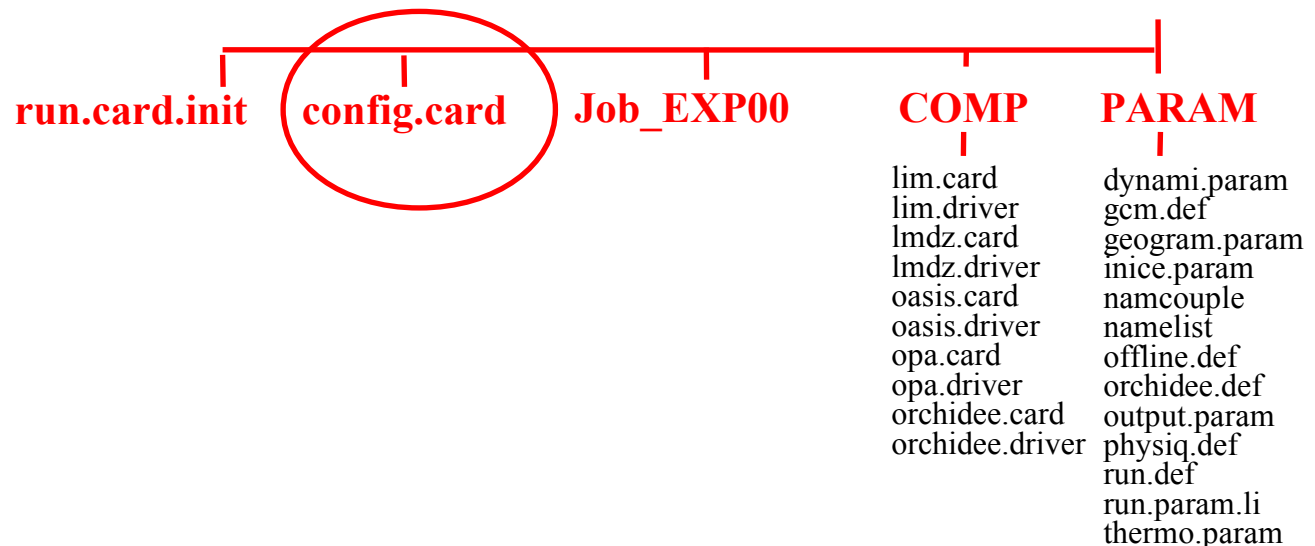
MY_EXPERIENCE

modipl

config

IPSLCM5A

EXP00



config.card : UserChoices

This is config.card file for IPSLCM5A configuration

##=====

#D-- Compatibility -

[Compatibility]

libIGCM=1.0

#D-- UserChoices -

[UserChoices]

#=====

JobName=**EXP00**

#----- Short Name of Experiment

ExperimentName=**pdControl**

#----- DEVT TEST PROD

SpaceName=**DEVT**

LongName="IPSLCM5A CMIP5 DEVT phase pdControl example with limited outputs"

TagName=**IPSLCM5A**

#=====

#-- leap, noleap, 360d

CalendarType=**noleap**

#-- Experiment dates : Beginning and ending

#-- "YYYY-MM-DD"

DateBegin=**2000-01-01**

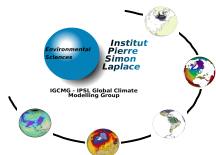
DateEnd=**2000-12-31**

#=====

#-- 1Y, 1M, 5D, 1D Period Length of one trunk of simulation

PeriodLength=**1M**

← Informations sur la simulation



config.card : Nb Processors

```
#=====
```

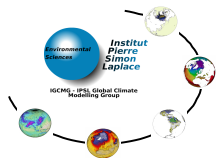
```
#-- Total Number of Processors (minimum is 2 for a coupled configuration)
```

```
JobNumProcTot=4
```

Note : 32 procs Titane ~ 4 procs Mercure en temps de calcul

Comment choisir le nombre de processus ? → La méthode de parallélisation de LMDZ impose la règle suivante : il faut au moins 3 bandes de latitude par processus
Si vous avez choisi un trop grand nombre de processus la simulation s'arrête avec le message suivant :

```
Arret : le nombre de bande de latitude par process est trop faible (<2).  
---> diminuez le nombre de CPU ou augmentez la taille en latitude
```



config.card : Composantes

```
#=====
```

```
#D-- ListOfComponents -
```

```
[ListOfComponents]
```

```
#D- For each component, Name of component, Tag of component
```

```
ATM= (lmdz, LMDZ4-dev)
```

```
SRF= (orchidee, ORCHIDEE_1_9_4)
```

```
OCE= (opa9, NEMO_v3_1 + revision 1340)
```

```
ICE= (lim2, LIM_2)
```

```
CPL= (oasis, OASIS3)
```

```
#=====
```

```
#D-- Executable -
```

```
[Executable]
```

```
Name=run_file
```

```
#D- For each component, Real name of executable, Name of executable for oasis
```

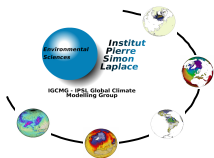
```
ATM= (gcm.e, lmdz.x)
```

```
SRF= ("", "")
```

```
OCE= (opa, opa.xx)
```

```
ICE= ("", "")
```

```
CPL= (oasis, oasis)
```



config.card : Restarts

```
#=====
#D-- Restarts -
[Restarts]
#D- If you want a GENERAL RULE FOR ALL COMPONENTS RESTARTS, put this flag to 'y'
OverRule=y
#D- Last day of the experience used as restart
RestartDate=1999-12-31
#D- Define restart simulation name
RestartJobName=EXP00
#D- Path Server Group Login
RestartPath=${ARCHIVE}/IGCM_OUT/IPSLCM5A/DEVT/pdControl
```

n pour un démarrage à zéro

Donner la date, le nom de la simulation
et le chemin d'accès aux fichiers

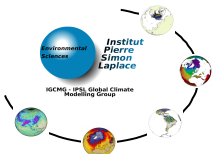
Dans cet exemple la simulation prendra comme fichiers de restart

Au ccrt :

```
/dmnfs/cont003/login/IGCM_OUT/IPSLCM5A/DEVT/pdControl/EXP00/...
/Restart/EXP00_...._19991231.nc
```

A l'Idris :

```
/u/rech/grp/login/IGCM_OUT/IPSLCM5A/DEVT/pdControl/EXP00/...
/Restart/EXP00_...._19991231.nc
```



config.card : une composante type ATM

```
#=====
```

```
#D-- ATM -
```

```
[ATM]
```

```
#
```

```
WriteFrequency="1M 1D HF"
```

```
# If config_Restarts_OverRule == 'n' all params are read
```

```
Restart= n
```

```
# Last day of the experience used as restart
```

```
RestartDate=1999-12-31
```

```
# Define restart simulation name
```

```
RestartJobName=EXP00
```

```
RestartPath=${ARCHIVE}/IGCM_OUT/IPSLCM5A/DEVT/pdControl
```

```
# Old component name for restart (if empty, use new name)
```

```
OldName=
```

y pour un redémarrage ATM
depuis une autre simulation

Donner la date, le nom de la simulation
et le chemin d'accès aux fichiers

```
#=====
```

```
#D-- OCE -
```

```
[OCE]
```

```
WriteFrequency="1M 1D"
```

```
Restart= n
```

```
###-- Last day of the experience used as restart
```

```
RestartDate=1999-12-31
```

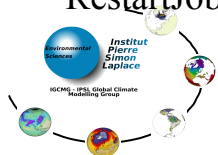
```
# Define restart simulation name
```

```
RestartJobName=EXP00
```

```
RestartPath=${ARCHIVE}/IGCM_OUT/IPSLCM5A/DEVT/pdControl
```

```
# Old component name for restart (if empty, use new name)
```

```
OldName=
```



config.card : Post

#-----

#D-- **Post** -

[Post]

#D- Do we rebuild parallel output, this flag determines

#D- frequency of rebuild submission (use NONE for DRYRUN=3)

RebuildFrequency=**2Y**

#D- Do we rebuild parallel output from archive

RebuildFromArchive=**none**

(true) Attention à l'espace disponible pour le stockage

#D- If you want to produce time series, this flag determines

#D- frequency of post-processing submission

TimeSeriesFrequency=**10Y (NONE)**

#D- If you want to produce seasonal average, this flag determines

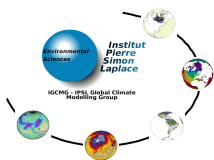
#D- the period of this average

SeasonalFrequency=**10Y (NONE)**

#D- Offset for seasonal average first start dates ; same unit as SeasonalFrequency

#D- Usefull if you do not want to consider the first X simulation's years

SeasonalFrequencyOffset=**0**



config.card : Vargas

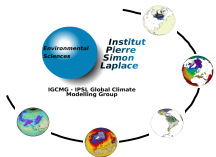
Il faut utiliser la bonne commande de lancement :

- Pour le couplé

```
JobRunOptions=' "-pgmmodel MPMD -cmdfile" '
```

- Pour les forcés

```
JobRunOptions=' '
```

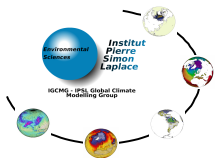


RebuildFromArchive

- A l'Idris nous conseillons le RebuildFromArchive
- Au CCRT nous conseillons le RebuildFromWorkdir → dans ce cas là il faut faire très attention aux quotas du scratchdir :

Pour 10 ans de simulation

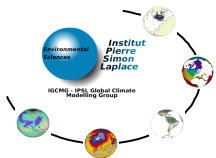
- historical de IPSLCM5A ~ 400 Go
- PiControl de IPSLCM5A ~ 210 Go
- donc toujours bien vérifier l'espace disque disponible
- réduire le RebuildFrequency en conséquence



Flux des données

Un fichier descriptif par composante (par ex opa9.card)

- Fichiers d'entrée texte (namelist)
- Fichiers d'entrée binaires :
 - conditions initiales
 - conditions limites
- Fichiers de sorties binaires (netCDF)
- Fichiers de sorties texte
- Fichiers de redémarrage



Flux de données

Les fichiers COMP/*.card

Les différentes rubriques :

```
[Compatibility]
```

```
LibIGCM=1.0
```

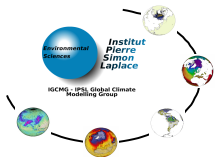
--> compatibilité avec la librairie libIGCM

```
[UserChoices]
```

```
[InitialStateFiles]
```

```
List= ($R_INIT}/SRF/${config_UserChoices_TagName}/soils_param.nc,      . ), \
      ($R_INIT}/SRF/${config_UserChoices_TagName}/routing.nc,          . ), \
      ($R_INIT}/SRF/${config_UserChoices_TagName}/PFTmap_IPCC_1850.nc,
PFTmap.nc)
```

--> fichiers d'états initiaux



Flux de données

les fichiers COMP/*.card

```
[BoundaryFiles]
```

```
List=(/dmnfs/cont003/p24data/ECMWF320x160/AN${year}/165_${year}${month}.nc,  
      u10mec.nc)  
ListNonDel=(${R_INIT}/CHM/LMDZORINCA/CH4/INCA${RESOL_CHM}/o3clim.nc,o3clim.nc) \  
            (${R_INIT}/CHM/LMDZORINCA/CH4/INCA${RESOL_CHM}/so4.nc ,so4.nc )
```

→ Liste des fichiers contenant les conditions aux limites. Deux catégories : List (fichiers rechargés à chaque période de simulation) et ListNonDel (chargés une seule fois)

```
[SmoothFiles]
```

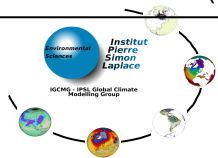
```
List=(${R_BC}/SRF/$  
      {config_UserChoices_TagName}/PFTmap_1850to2005_AR5_LUHa.rc2/PFTmap_IPCC_$  
      {year}.nc, PFTmap.nc, 12:12:)
```

→ Liste de fichiers rechargés toutes les n-périodes de simulations

```
[ParametersFiles]
```

```
List= (${SUBMIT_DIR}/PARAM/orchidee.def, .)
```

→ Liste des fichiers de paramètres (normalement stockés dans PARAM/)



Flux de données

Les fichiers COMP/*.card

```
[RestartFiles]
```

```
List=(${config_UserChoices_JobName}_${NEMO_END}_restart.nc, restart.nc,  
      restartopa.nc)
```

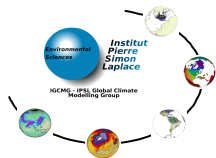
```
[RestartFiles]
```

```
List=(restart.nc, restart.nc, start.nc), \  
      (restartphy.nc, restartphy.nc, startphy.nc)
```

→ Liste des fichiers de restarts pour le chainage des simulations

Syntaxe :

```
List= (restart en sortie du modèle, restart stocké, restart en entrée du  
      modèle pour le chainage)
```



Flux de données

Les fichiers COMP/*.card

```
[OutputText]  
List= (physiq.def, run.def)
```

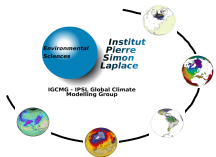
→ Liste des fichiers textes que l'on veut conserver en fin de simulation

```
[OutputFiles]  
List= (sechiba_history.nc, ${R_OUT_SRF_O_M}/${  
    {PREFIX}_1M_sechiba_history.nc, Post_1M_sechiba_history),  
    (orchidee_watchout.nc, ${R_OUT_SRF_O_M}/${  
    {PREFIX}_1M_watchout.nc, NONE)
```

→ Liste des fichiers output

Syntaxe :

```
List= (output file, stockage, post-traitement appliqué)
```



Flux de données

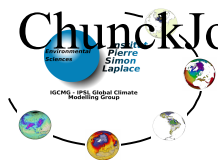
Fichiers COMP/*.card

```
[Post_1M_sechiba_history]
Patches= ()
GatherWithInternal= (lon, lat, veget, time_counter, time_counter_bnds, Areas,
    Contfrac)
TimeSeriesVars2D= (nobiofrac, alb_nir, alb_vis, bqsb, evap, fluxlat, fluxsens,
    gqsb, netrad, qair, rain, runoff, snow, snownobio, snowf, subli, tair,
    temp_sol, tsol_max, tsol_min, drainage, mrsos, mrso, mrros, mrro, prveg,
    evspsblveg, evspsblsoi, tran, treeFrac, grassFrac, cropFrac, baresoilFrac,
    residualFrac)
ChunckJob2D= NONE
TimeSeriesVars3D= (lai, maxvegetfrac, vegetfrac, CO2FLUX, ptn, nee)
ChunckJob3D= NONE
Seasonal= ON
```

→ Liste des post-traitements à appliquer aux fichiers d'Output

GatherWithInternal = liste des variables à ajouter aux times series

ChunckJob2D/3D = NONE si time series sur toute la simul,
= 50Y si time series par tranche de 50Y



Fichiers COMP/* .card

Variables et syntaxe des fichiers COMP/* .card :

Au ccrt :

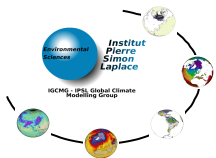
$\${R_INIT}$ = /dmnfs/cont003/p86ips1/IGCM/INIT/

$\${R_BC}$ = /dmnfs/cont003/p86ips1/IGCM/BC/

A l'Idris

$\${R_INIT}$ = /u/rech/psl/rpsl035/IGCM/INIT/

$\${R_BC}$ = /u/rech/psl/rpsl035/IGCM/BC/



Fichiers COMP/* .card

Sur toutes les machines :

$\{\text{SUBMIT_DIR}\}$ = répertoire d'expérience

$\{\text{year}\}$ = année en cours

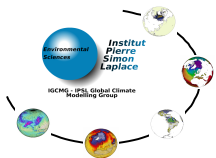
$\{\text{month}\}$ = mois en cours

$\{\text{config_UserChoices_JobName}\}$ = nom du Job (JobName dans config.card)

(path/filename, newfilename) = cp path/filename newfilename

(..., ...) , \ = la liste continue à la ligne suivante

Attention : ne pas mettre d'espace en fin de ligne



Les monitorings

Modifier/créer un fichier de type EXP_../POST/monitoring01_model.cfg

```
field | files patterns | files additionnal | operations | title | units | calcul of area
#-----
NOX_surf_global | "NO NO2" | LMDZ4.0_9695_grid.nc | "(NO[d=1,k=19]+NO2[d=2,k=19])" | " NOX
a la surface" | "VMR" | "aire[d=3]"

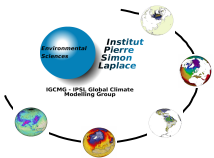
CH4_surf_global | "CH4" | LMDZ4.0_9695_grid.nc | "CH4[d=1,k=19]" | "
CH4 a la surface" | "VMR" | "aire[d=2]"

CO_surf_global | "CO" | LMDZ4.0_9695_grid.nc | "CO[d=1,k=19]" | " CO
a la surface" | "VMR" | "aire[d=2]"

O3_surf_global | "_O3" | LMDZ4.0_9695_grid.nc | "O3[d=1,k=19]" | " O3
a la surface" | "VMR" | "aire[d=2]"

HNO3_surf_global | "HNO3" | LMDZ4.0_9695_grid.nc | "HNO3[d=1,k=19]"
```

Attention : il faut que les variables que l'on veut visualiser dans les monitoring soient sorties en time series



Un peu plus en détail...

Répertoire EXP00 prêt :

- COMP/* : information sur les composantes
- config.card : fichier de configuration de la simulation
- Job_EXP00 : Job à soumettre
- PARAM/* : fichiers de configuration des modèles
- run.card.init : fichier de suivi original

MY_EXPERIENCE

modipl

config

IPSLCM5A

EXP00

run.card.init

config.card

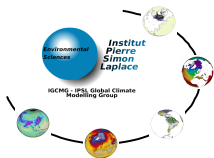
Job_EXP00

COMP

lim.card
lim.driver
lmdz.card
lmdz.driver
oasis.card
oasis.driver
opa.card
opa.driver
orchidee.card
orchidee.driver

PARAM

dynami.param
gcm.def
geogram.param
inice.param
namcouple
namelist
offline.def
orchidee.def
output.param
physiq.def
run.def
run.param.li
thermo.param



Job_xxx

```
#!/usr/bin/ksh
#####
## TITANE CEA ##
#####
#MSUB -r LOINA      # nom de la requete
#MSUB -o Script_Output_LOINA.000001  # nom du fichier de sortie
#MSUB -n 32         # reservation des processeurs pour le job
#MSUB -T 86400      # Limite temps (en secondes)
#MSUB -p gen2211
BATCH_NUM_PROC_TOT=$BRIDGE_MSUB_NPROC
```

#D- Increased verbosity (1, 2, 3)

Verbosity=3

#D- Number of execution in one job

PeriodNb=1

#D- Experience type : DEB(ug), DEV(elopment), RUN (default)

JobType=RUN

#D- Define running directory

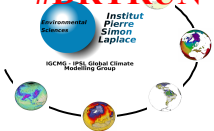
#D- Default=\${TMPDIR} ie temporary batch directory

#RUN_DIR_PATH=\$SCRATCHDIR

#D- Turn in dry run mode ? (0,1,2,3)

#D- Default=0

#DRYRUN=3



```
# YOU MUST COMPILE YOUR EXE FILES FOR DRYRUN MODE !
# -----
# | DRYRUN= | Date computations, | sys_Get | Exe      | sys_Put_Out; sys_Put_Res
# |         | Cp/Exe param files |         | Chmod    |
# |         | Qsub                |         |          |
# -----
# | 0       | yes                 | yes     | yes      | yes      |
# -----
# | 1       | yes                 | yes     | yes      | no       |
# -----
# | 2       | yes                 | yes     | no       | no       |
# -----
# | 3       | yes                 | no      | no       | no       |
# -----
```

Job_xxx (titane) :

choix du groupe de soumission

Attention le groupe genci est défini par défaut dans l'entête du job.

→ se référer à sa demande d'heures (voir avec son responsable éventuellement)

Heures genci : modifier la ligne `#MSUB -p gen2211` dans l'entête du Job_xxx en indiquant son groupe (commande groups). Faire de même dans l'entête du job de rebuild : `modips1/libIGCM/rebuildfromWorkdir.job`

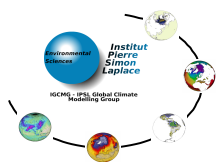
Heures dsm : commenter la ligne `#MSUB -p gen2211` dans l'entête du Job_xxx ainsi que dans l'entête du job de rebuild : `modips1/libIGCM/rebuildfromWorkdir.job`. Attention dans ce job il faut également modifier la queue de soumission :

```
#!/bin/ksh
#####
## TITANE CEA ##
#####
#MSUB -r REBUILDWRK
#MSUB -eo
#MSUB -n 1
#MSUB -T 86400
#####MSUB -p gen2201
#MSUB -q monoext
```

Remarques :

1- Pour conserver les modifications vous pouvez modifier directement le fichier `libIGCM/AA_job` et refaire la commande `ins_job`

2- La commande `class` vous permet de connaître les queues existantes sur titane. D'une manière générale pour les heures dsm ce sont les queues « ext » et pour les heures genci les autres.



Job_xxx : PeriodNb

Lancement de plusieurs périodes par job

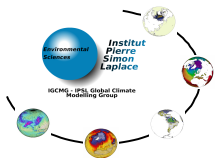
Pour éviter de lancer une foule de petits jobs qui reprennent la file d'attente à chaque fois, il est possible de lancer en boucle n périodes (PeriodLength) par job.

Le paramètre à modifier est dans *Job_JobName* (1 par défaut) :

PeriodNb=1

Attention! Modifier le paramètre du temps dans l'entête en conséquence.

Pour connaître les différentes queues disponibles sur une machine il faut utiliser la commande `class` au CCRT et `news class` à l'IDRIS.



Soumission - contrôle

MY_EXPERIENCE

- **qsub Job_EXP00**
- **Contrôle**
 - Mercure : qstat, mpp
 - Brodie : qstat
- Répertoire EXP00 en cours de simulation :
 - COMP/*
 - PARAM/*
 - run.card.init
 - **run.card**
 - config.card
 - Job_EXP00
 - **Script_Output***

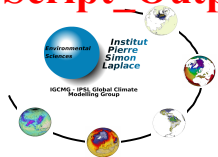
|
modips1

|
config

|
IPSLCM5A

|
EXP00

Script_Output* run.card run.card.init config.card Job_EXP00 COMP PARAM

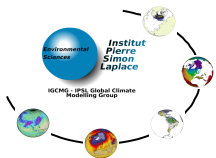


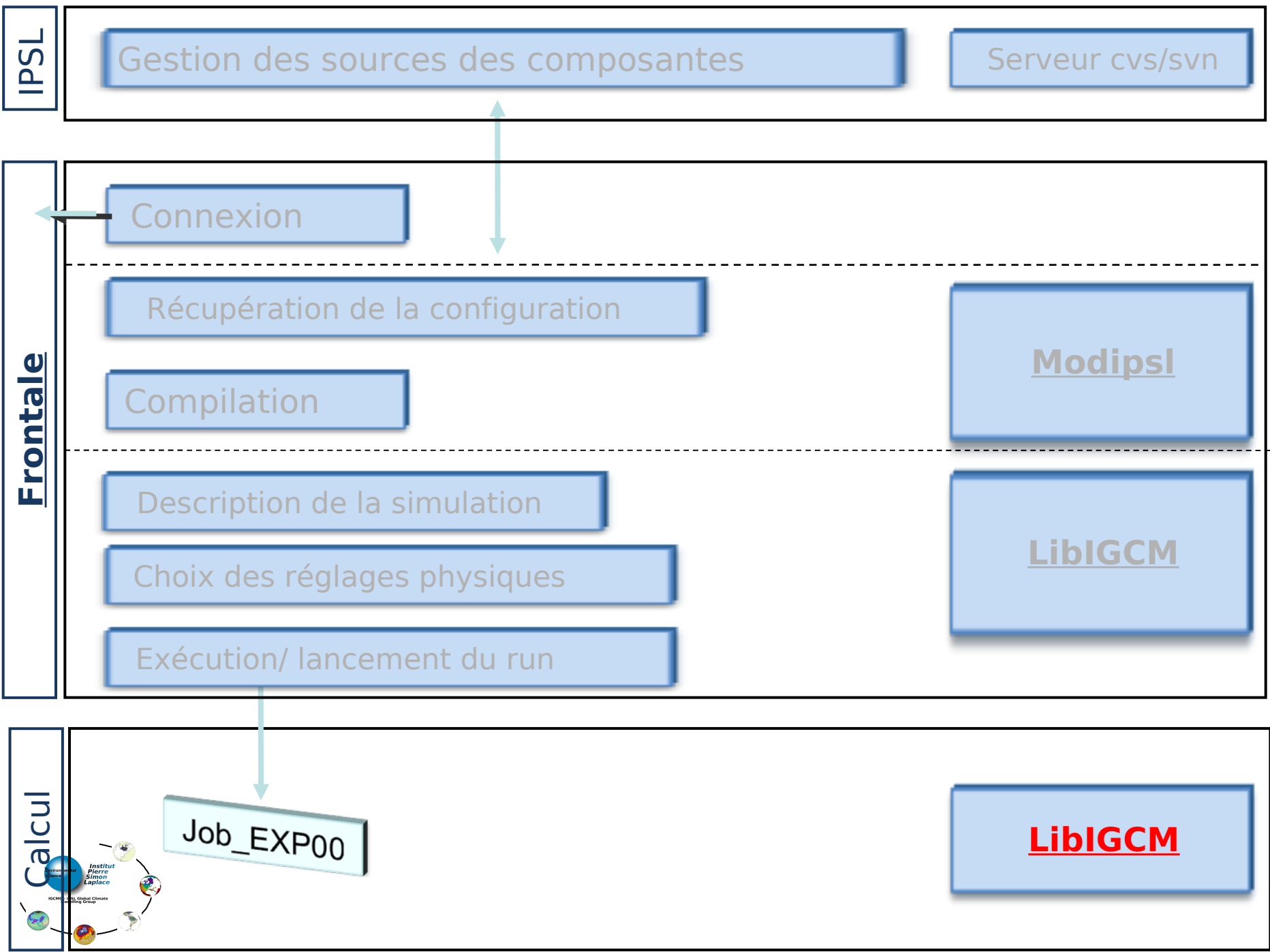
run.card : le fichier de suivi

```
# contient la date de la periode en cours ou en attente
# last date of loop == .suivi
[Configuration]
#last PREFIX
OldPrefix= EXP00_20000131
#Compute date of loop == .suivi
PeriodDateBegin= 2000-02-01
PeriodDateEnd= 2000-02-28
CumulPeriod= 2
# State of Job "Start", "Running", "OnQueue", "Completed"
PeriodState= Running

[PostProcessing]

TimeSeriesRunning=n
TimeSeriesCompleted=
```





IPSL

Gestion des sources des composantes

Serveur cvs/svn

Frontale

Connexion

Récupération de la configuration

Compilation

Modipsl

Description de la simulation

Choix des réglages physiques

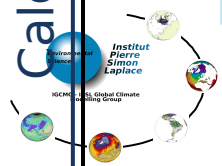
Exécution/ lancement du run

LibIGCM

Calcul

Job_EXP00

LibIGCM



Calcul

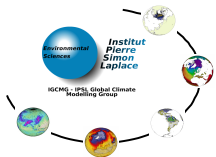
Job_EXP00



Job_EXP00



LibIGCM



Calcul

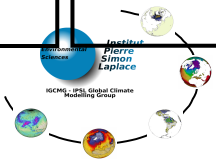


LibIGCM

Machine de post-traitement

Traitement des fichiers de sorties

LibIGCM



Calcul



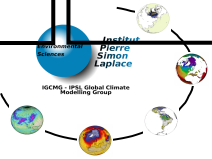
LibIGCM

Machine de post-traitement



Outil de recombinaison des fichiers produits par chaque processus de calcul. Etape en mode asynchrone.

LibIGCM

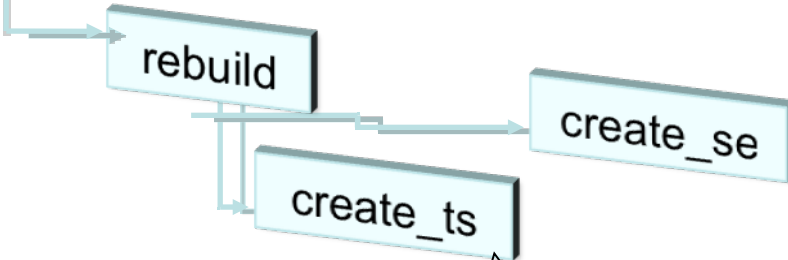


Calcul



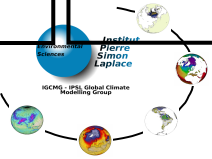
LibIGCM

Machine de post-traitement



Génération de séries temporelles de variables spécifiques.

LibIGCM

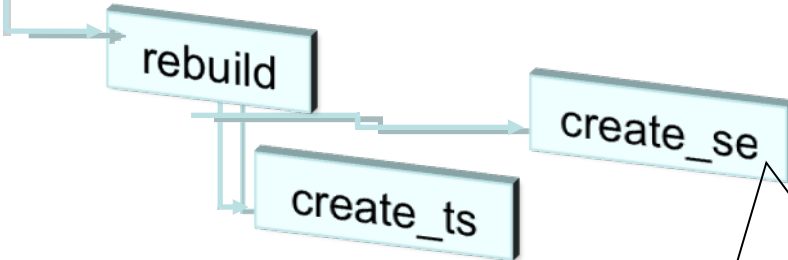


Calcul



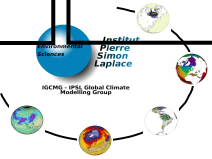
LibIGCM

Machine de post-traitement



Génération de moyennes saisonnières.

LibIGCM



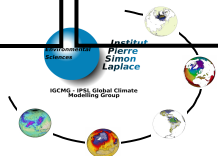
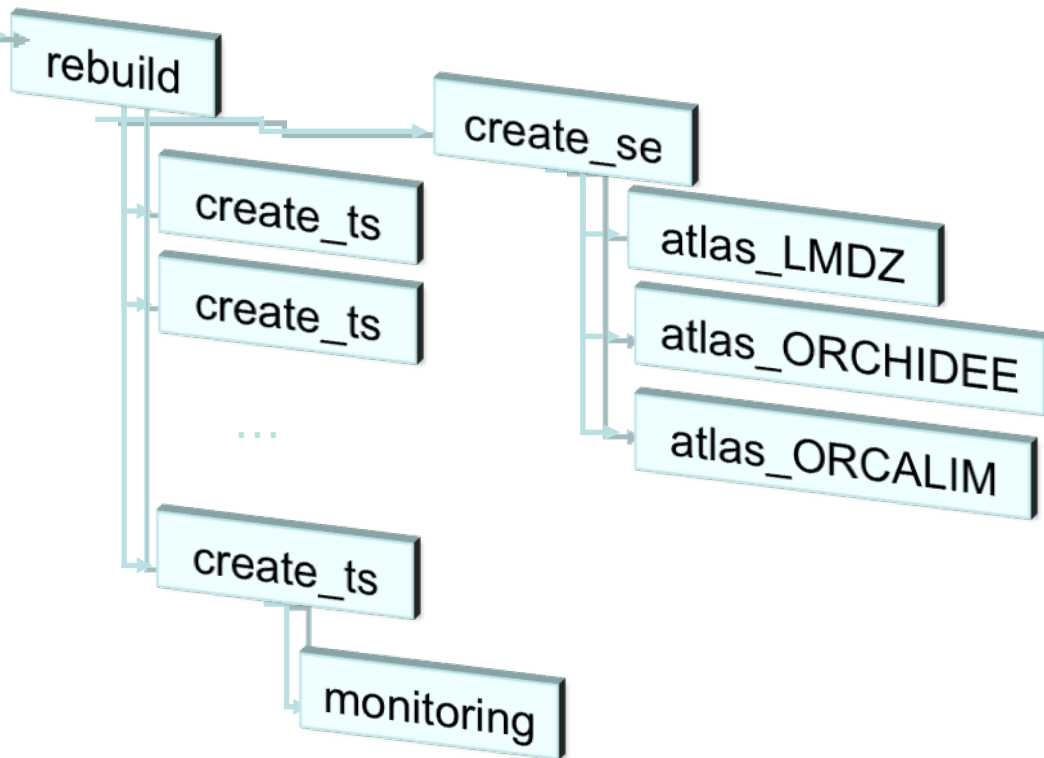
Calcul

LibIGCM



Machine de post-traitement

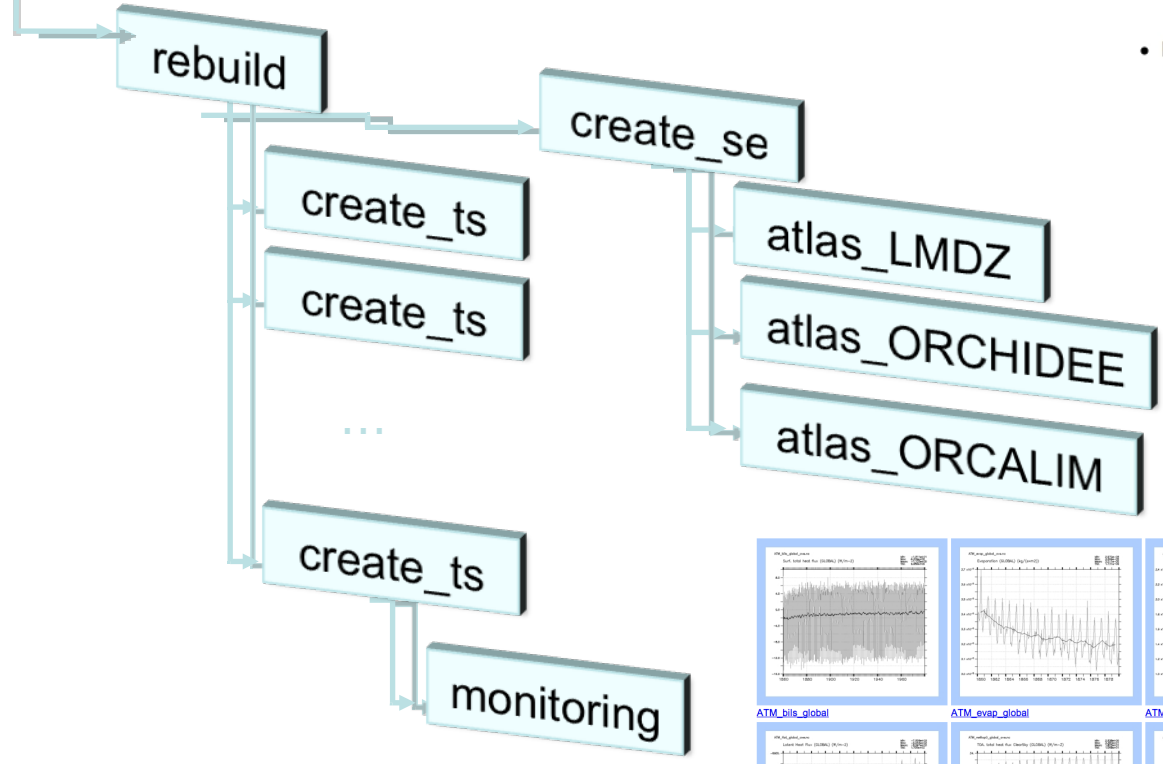
LibIGCM



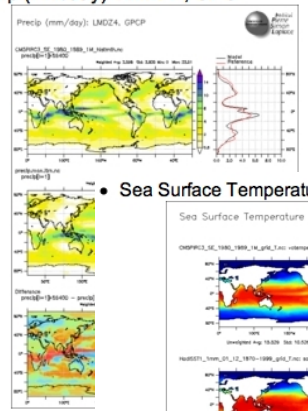
Calcul



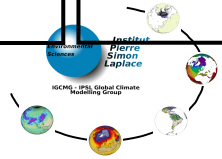
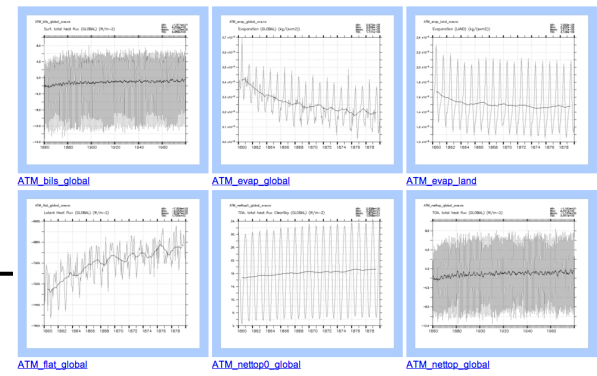
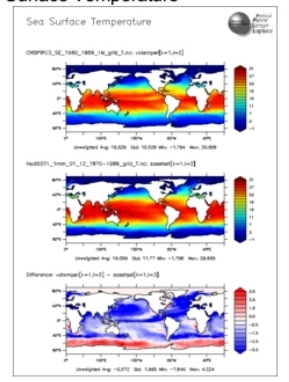
Machine de post-traitement



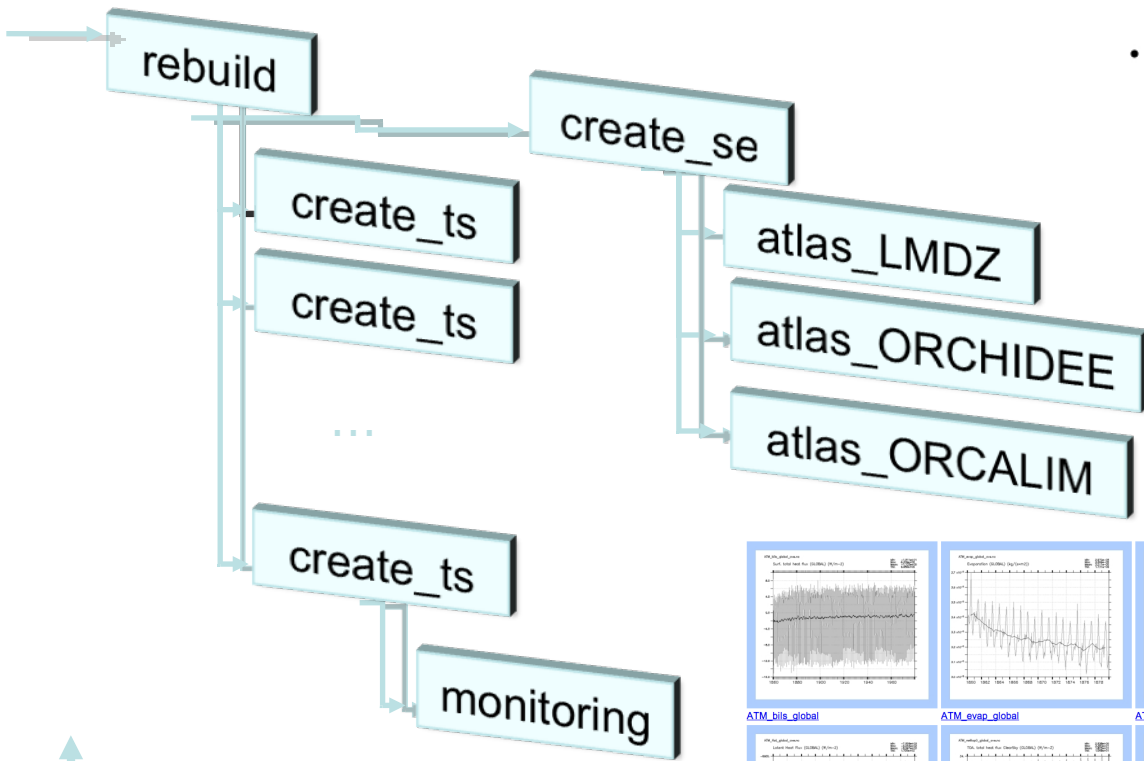
• Precip (mm/day): LMDZ4, GPCP



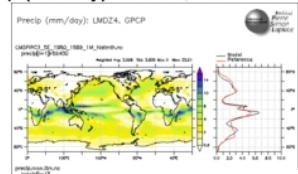
• Sea Surface Temperature



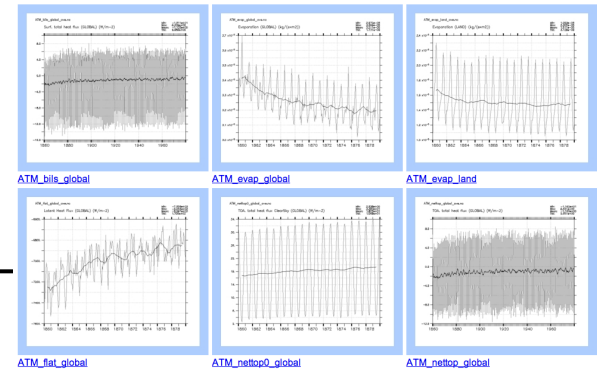
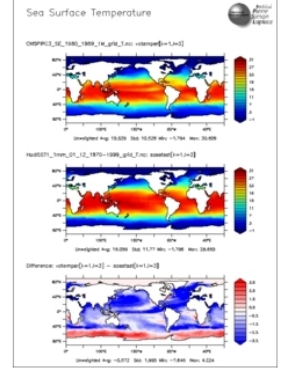
Machine de post-traitement



• Precip (mm/day): LMDZ4, GPCP



• Sea Surface Temperature

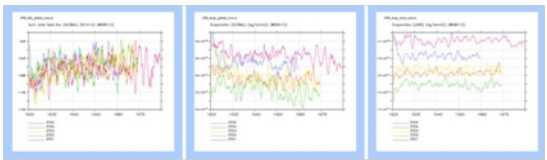


Archive

Fichiers bruts

Fichiers post-traités et analyses

Web



Fichiers post-traités et analyses

Les utilitaires de post-traitement CCRT, IDRIS

create_ts.job : séries temporelles tous les 10 ans

create_se.job : moyennes saisonnières tous les 10 ans

Retour des jobs de post-traitement là :

ulam : \$WORKDIR/IGCM_OUT/IPSLCM5A/*JobName*

Mercure/cesium : \$SCRATCHDIR/IGCM_OUT/IPSLCM5A/*JobName*

atlas_ORCA_LIM : pour océan et glace de mer

atlas_LMDZ : pour atmosphère

atlas_ORCHIDEE : pour surfaces continentales

Les atlas sont basés sur *ferret* et sur *fast* :

<http://dods.ipsl.jussieu.fr/fast/>

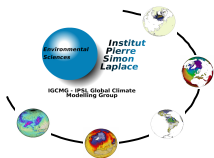
<http://wiki.ipsl.jussieu.fr/IGCMG/Outils/ferret/60MinutesAvecFerret>

MY_EXPERIENCE

modipsl

libIGCM

- **create_ts**
- **create_se**
- **atlas_...**
- **monitoring**
- **clean_month**



Message en fin de simulation

A la fin d'une expérience, vous recevrez un message de ce type :

Dear *login*,

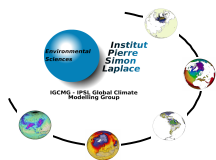
```
Simulation EXP00 is finished on supercomputer brodie03.
```

```
Job started : 20000101
```

```
Job ended   : 20001231
```

```
Ouput files are available in
```

```
.../IGCM_OUT/IPSLCM5A/DEVT/pdControl/EXP00
```



Comment vérifier que cela s'est bien passé?

- **run.card** :
PeriodState=Completed
- Message de fin de simu reçu
- Fichiers sur le serveur de fichiers
- Post-traitements lancés puis finis
- ATLAS et monitoring sur serveur dods

MY_EXPERIENCE

|
modipsl

|
config

|
IPSLCM5A

|
EXP00

Script_Output

run.card

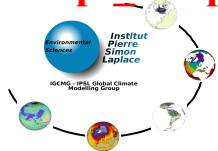
run.card.init

config.card

Job_EXP00

COMP

PARAM

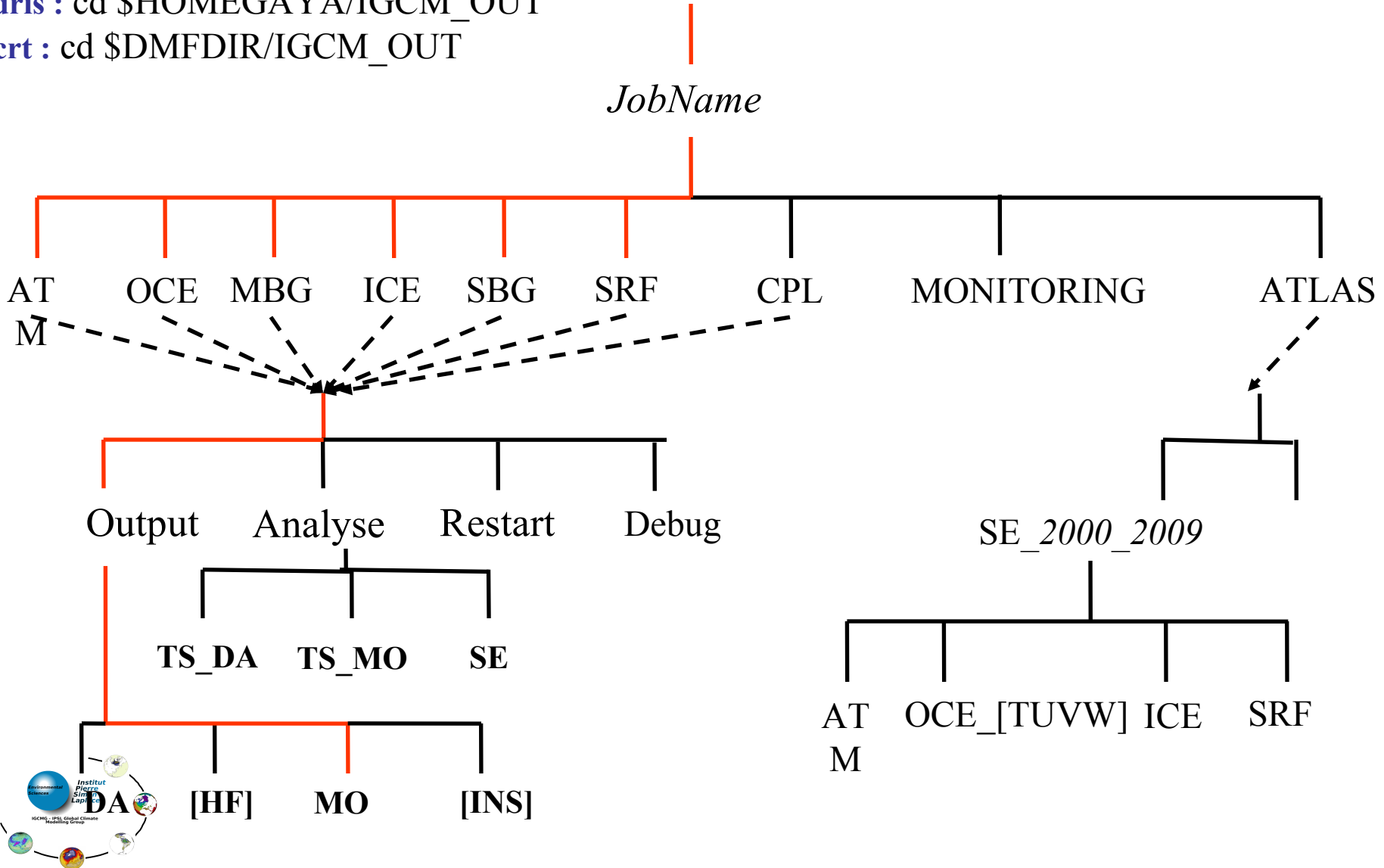


Arborescence sur serveur fichiers

IPSLCM5A/DEVT/pdControl

Idris : cd \$HOMEGAYA/IGCM_OUT

ccrt : cd \$DMFDIR/IGCM_OUT



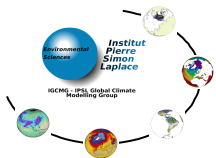
Accès aux résultats de simulations

- Mise en ligne du monitoring et des atlas sur les serveurs dods :

dods IDRIS : <http://dods.idris.fr/monlogin>

dods CCRT : <http://dods.extra.cea.fr/data/monlogin>

- Accès aux simulations de référence
 - Serveur Prodiguer : distributions résultats simulations CMIP5 IPSL
 - Ciclad : serveur calcul données à l'IPSL
<http://ciclad-web.ipsl.jussieu.fr/>



Nomenclature des noms des fichiers de sortie

Output, Analyse, Debug, ...

$\{\text{JobName}\}_{_}\{\text{PeriodDateBegin}\}_{_}\{\text{PeriodDateEnd}\}_{_}\text{XX}_{_}\text{NomFichier}$

Output/**DA** et Analyse/TS_**DA**:

XX \longrightarrow **1D**

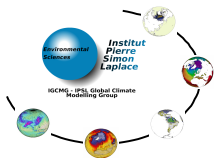
Output/**MO** et Analyse/TS_**MO**

XX \longrightarrow **1M**

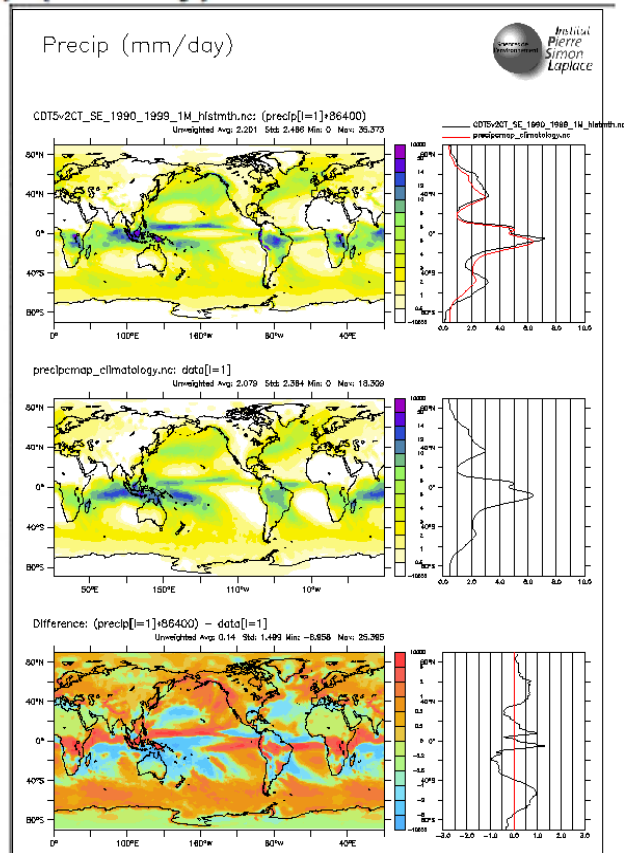
Analyse/**SE** :

$\{\text{JobName}\}_{_}\text{SE}_{_}\{\text{PeriodDateBegin}\}_{_}\{\text{PeriodDateEnd}\}_{_}\text{NomFichier}$

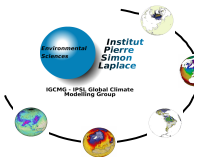
Restart : $\{\text{JobName}\}_{_}\{\text{PeriodDateEnd}\}_{_}\text{NomFichier}$



- Atm
 - [Atm.pdf](#) (25328 Ko)
- Precip (mm/day)



- [precip_13185/precip.gif](#) (608 Ko)
- [precip_13185/precip.pdf](#) (1088 Ko)



pdControl_EXP00 monitoring

at 2010-04-22 17:08:37

Cards Analysis Monitoring Board About

ALL

Filter : *

Images : 077 / 077

ATM

CHM

ICE

MBG

OCE

SBG

SRF

XOR

CLR

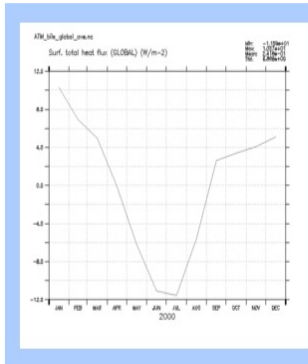
land

ocean

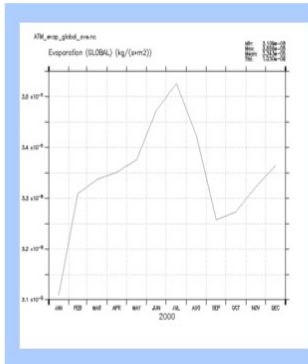
north

south

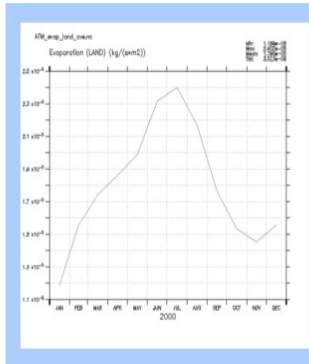
global



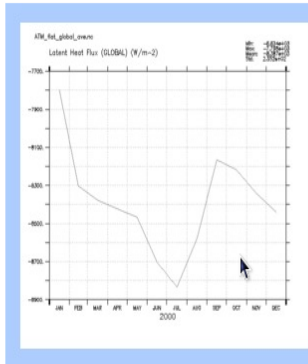
ATM_bils_global



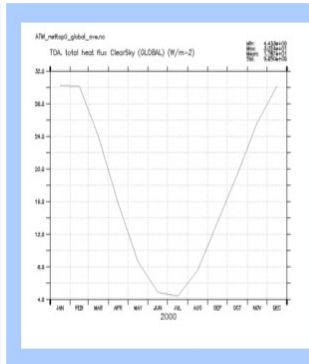
ATM_evap_global



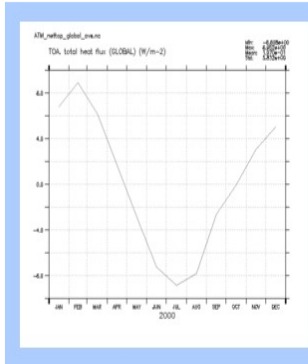
ATM_evap_land



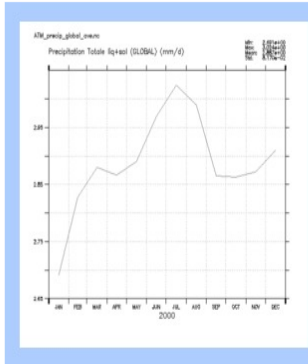
ATM_flat_global



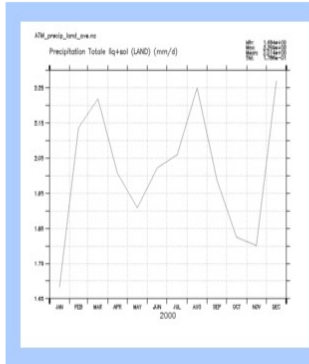
ATM_nettop0_global



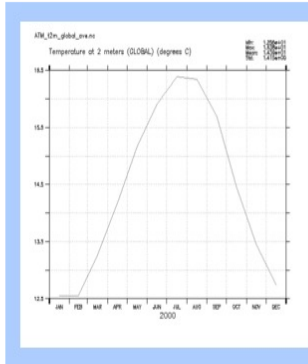
ATM_nettop_global



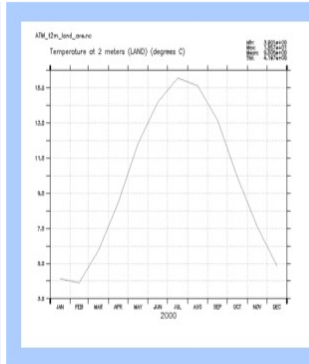
ATM_precip_global



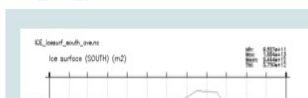
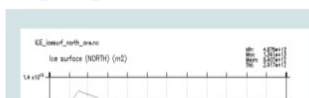
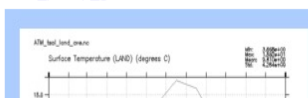
ATM_precip_land



ATM_t2m_global



ATM_t2m_land



v2.historical1 monitoring

at 2010-10-30 20:10:25

[Cards Analysis](#) [Monitoring Board](#) [About](#)

ALL

Filter : *

Images : 067 / 067

ATM

CHM

ICE

MBG

OCE

SBG

SRF

XOR

CLR

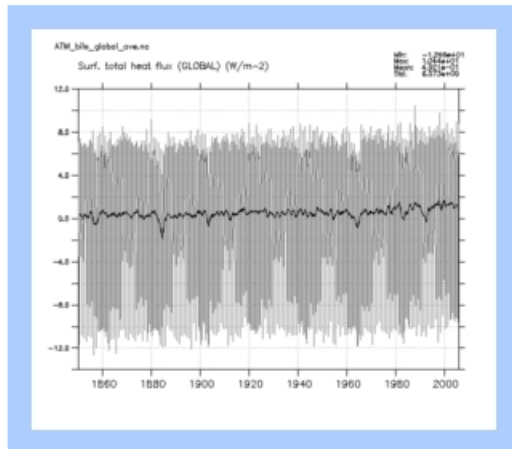
land

ocean

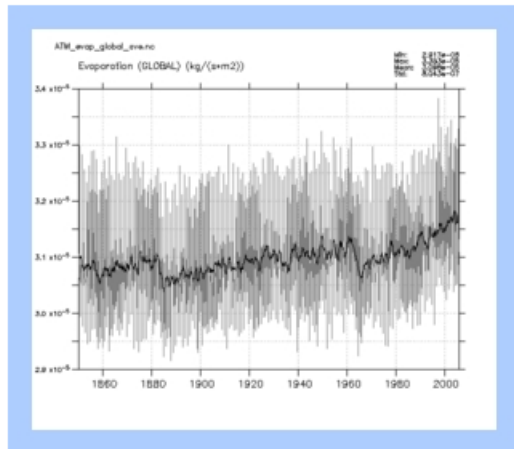
north

south

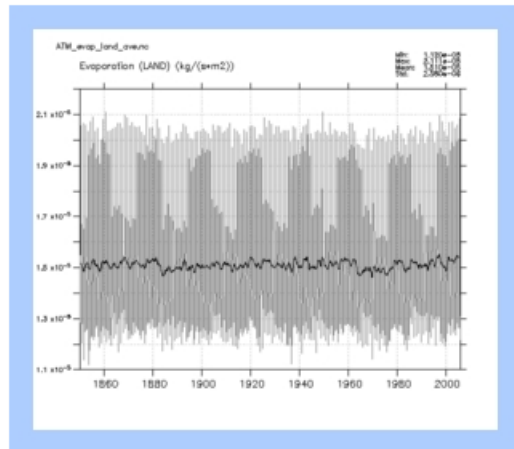
global



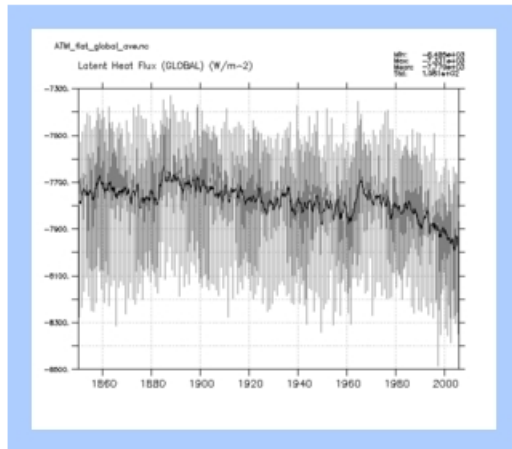
[ATM_bils_global](#)



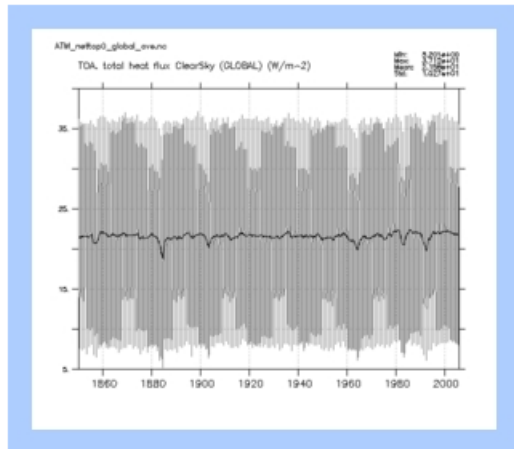
[ATM_evap_global](#)



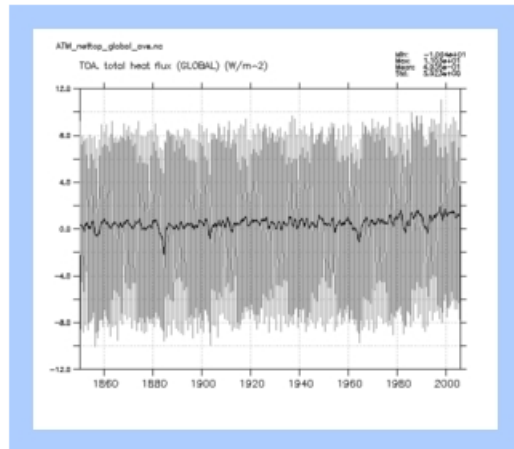
[ATM_evap_land](#)



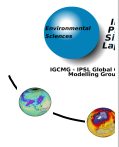
[ATM_flat_global](#)



[ATM_nettop0_global](#)



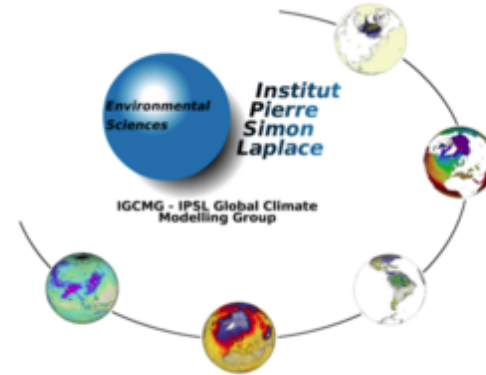
[ATM_nettop_global](#)








IGCMG Web services

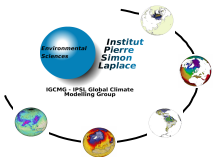
<http://igcmg.lsce.ipsl.fr/>

Welcome to the IGCMG Web Services portal



Services

-  [Trusting Web Service](#)
-  [Machine Load Status](#)
-  [Inter Monitoring Web Service](#)
-  [Meta Atlas Web Service](#)
-  [Metrics Web Service](#)

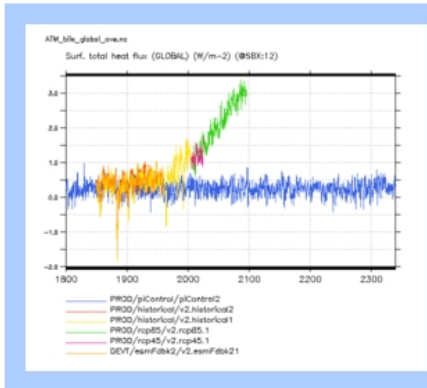


Monitoring comparison: piControl2 vs v2.historical2 vs v2.historical1 vs v2.rcp85.1 vs v2.rcp45.1 vs v2.esmFdbk21

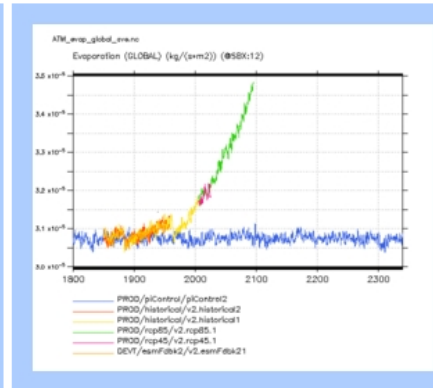
at 2010-11-17 17:15:26

ALL Filter : . Images : 064 / 064

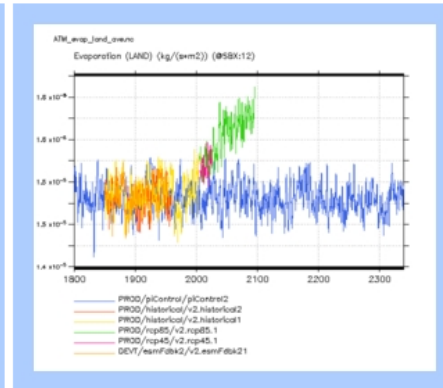
ATM	CHM	ICE	MBG	OCE	SBG	SRF	XOR	CLR
land	ocean	north	south	global				



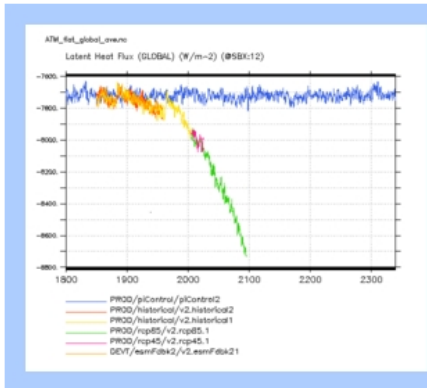
[ATM_bils_global_ave](#)



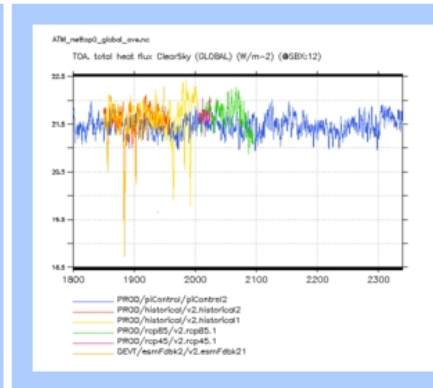
[ATM_evap_global_ave](#)



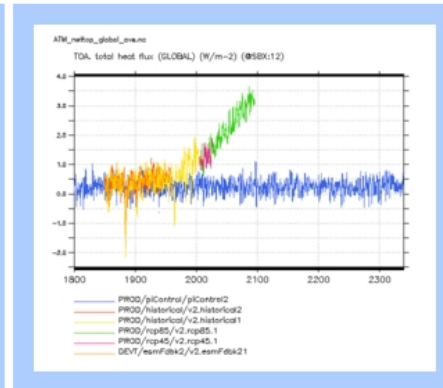
[ATM_evap_land_ave](#)



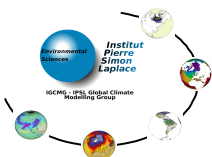
[ATM_flat_global_ave](#)



[ATM_nettop0_global_ave](#)



[ATM_nettop_global_ave](#)



InterMonitoring – tutoriel

Inter Monitoring Web Service Release: 1.20

Monitor your different simulations run with the libIGCM environment production. [Ask for support](#)
[Discover other services](#)

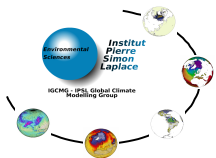
Step 1 | Step 2 | Step 3 | Step 4

Enter URL to find simulation output directories

Recall of your choices

- <http://dods.extra.cea.fr/cgi-bin/nph-dods/data/p24diana/IPSLCM5A/PROD/rcp45v3.rcp45.1>, <http://dods.extra.cea.fr/cgi-bin/nph-dods/data/p86denv/IPSLCM5A/PROD/rcp45v3.rcp45.1>
- ATM_absvisae forcing_ave.nc
- plot01
- Smoothing box value:
- Dates default
- Dates range: 1800 - 2100

Vidéo du tutoriel : <http://dods.ipsl.jussieu.fr/brocksce/screencast/InterMonitoring.html>

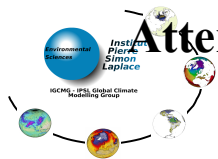


Questions fréquentes sur les simulations

- Comment écraser une simulation existante ?
 - Effacer le fichier run.card
 - Effacer le répertoire d'output sur le disque de stockage (DMFDIR/gaya)
 - Soumettre
- Comment continuer une simulation ?
 - Mettre à jour config.card, vérifier que le fichier run.card est prêt pour la période suivante

Ex: on vient de finir le mois de janvier 2001 et on se prépare pour février

```
OldPrefix= EXP00_20000131  
PeriodDateBegin= 2000-02-01  
PeriodDateEnd= 2000-02-28  
PeriodState=OnQueue
```



Attention : si vous continuez une simulation qui a buggué, il faut vérifier qu'elle n'a pas créé de fichiers pour le mois que vous allez relancer (utilisation du script `clean_month` dans `libIGCM`)

Comment savoir où est le problème?

Si dans **run.card** : `PeriodState=Fatal`

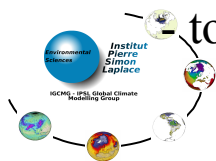
--> Regarder le fichier **Script_Outputxxxx**

```
#####  
#          ANOTHER GREAT SIMULATION          #  
#####  
1ère partie  
#####  
#          DIR BEFORE RUN EXECUTION          #  
#####  
2ème partie  
#####  
#          DIR AFTER RUN EXECUTION          #  
#####  
3ème partie
```

Erreurs dans la première partie - différentes pistes:

- si on commence une nouvelle simulation : est-ce qu'il y avait ou non un `run.card` ?
- si on continue une simulation : a-t-on bien mis `PeriodState=OnQueue` dans le `run.card`

tous les fichiers d'input existent-ils ?



```

--Debug1--> !!!! IGCM_config_Initialize Error PeriodState :
--Debug1--> Completed
--Debug1--> !!!!!!!!!!!!!
IGCM_debug_Exit : IGCM_config_Initialize Error PeriodState : xxxx
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! IGCM_debug_CallStack !!
!-----!
IGCM_debug_Verif_Exit : Something wrong append.
                        EXIT THE JOB.
Fri Apr 22 12:14:29 MET 2011

```

Il y avait déjà un run.card
avec une PeriodState ≠ de
OnQueue
xxxx = Completed ou Fatal

```

--Debug1--> IGCM_post_CheckModuloFrequency : Master=1M Slave=1D
-----Debug3--> config_UserChoices_PeriodLength frequency 1D
not compatible with
-----Debug3--> config_Post_RebuildFrequency frequency : 1M
IGCM_debug_Exit : Check your frequency

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! IGCM_debug_CallStack !!
!-----!
--Debug1--> IGCM_post_CheckModuloFrequency : Master=1M Slave=1M
--Debug1--> IGCM_post_CheckModuloFrequency : Master=1M Slave=1M
--Error--> IGCM_card_WriteOption
/scratch/cont003/p24cozic/RONA/modipsl/config/LMDZINCA_v3/EXP_AER_2/run
.card Configuration PeriodState Fatal

/scratch/cont003/p24cozic/RONA/modipsl/config/LMDZINCA_v3/EXP_AER_2/run
.card is not readable or not writable
IGCM_debug_Exit : IGCM_card_WriteOption
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! IGCM_debug_CallStack !!
!-----!
IGCM_debug_Verif_Exit : Something wrong append.
                        EXIT THE JOB.

```

Les fréquences de
post-traitements ne
sont pas bonnes



Comment savoir où est le problème ?

Erreurs dans la deuxième partie :

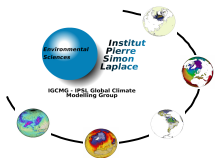
- a-t-on demandé assez de temps d'exécution ?
- a-t-on demandé assez de mémoire ?
- si on a le message suivant :

```
=====
EXECUTION of : mpirun -f ./run_file > out_run_file 2>&1
Return code of executable : 1
IGCM_debug_Exit : EXECUTABLE

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!! IGCM_debug_CallStack !!
!-----!

!-----!
IGCM_sys_Cp : out_run_file xxxxxxxxxxxx_out_run_file_error
=====
```

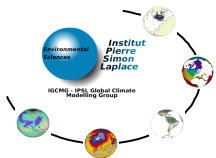
--> analyser le fichier xxxxxxxxxxxx_out_run_file_error



Comment savoir où est le problème ?

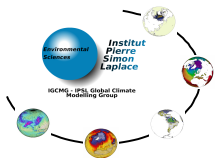
Erreurs dans la troisième partie :

- Si on écrase une simulation a-t-on bien effacé les fichiers de sorties de la simulation d'origine ?
- Si il n'y a pas de fichier de restart, vérifier qu'il n'y ait pas une erreur dans la deuxième partie
 - Si il manque un fichier de restart et pourtant vous n'avez pas de message d'erreur dans la seconde partie vous êtes peut être tombés sur un garde-fou (dans LMDZ par ex). Il faut alors relancer la simulation en indiquant `RUN_DIR_PATH=$SCRATCHDIR` dans `Job_jobname` et ensuite analyser le fichier `out_gcm.e` dans le répertoire créé dans le `$SCRATCHDIR`.
- Tout s'est bien passé mais les post-traitements n'ont pas fonctionné --> relancer les post-traitements



Comment savoir où est le problème ?

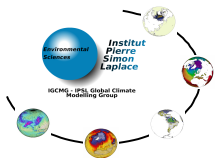
- Regarder le fichier **xxxxx_error** dans le répertoire de soumission
 - Contient le texte de sortie de LMDZ
LMDZ s'arrête souvent dans hgardfou
Stopping in hgardfou
 - Contient les **erreurs brutales** de toutes les composantes
- Regarder le fichier texte de sortie de NEMO
 - Serveur de fichiers
 - **EXP00/OCE/Debug/EXP00_xxxxxx_ocean.output**
 - Les erreurs NEMO sont synthétisées à la fin.
- Regarder les fichiers texte de sortie de ORCHIDEE
 - Serveur de fichiers
 - **EXP00/SRF/Debug/EXP00_xxxxxx_output_orchidee_0000**
, ... _0003



Lancer les post-traitements à postériori

Remarque : Tous les fichiers devant être reconstruits sont sauvegardés

- Si `RebuildFromArchive=.true.` : dans `IGCM_OUT/.../EXP00/TMP/`
 - Si `RebuildFromArchive=NONE` : sur le `$$SCRATCHDIR` au `ccrt` et sur le `$WORKDIR` à l'idris (attention aux quotas limités)
-
- Si ce sont les rebuilds qui n'ont pas fonctionné vous pouvez les relancer et ils lanceront automatiquement les TS et SE en découlant
 - Si ce sont les TS ou les SE vous pouvez les relancer sans passer par les rebuilds.
 - Vous pouvez également utiliser l'outil `TimeSeries_Checker` qui vérifie les séries temporelles existantes et relance les jobs `create_TS` nécessaires pour reconstruire les TS manquantes.
- > toutes les explications sont sur la page `ModiplBeginner`



Le fichier xxx_out_run_file_error

- Ce fichier contient les prints des modèles LMDZ et INCA (si la simulation se fini normalement vous les retrouvez dans le répertoire :
\$DMFDIR/IGCM_OUT/.../my_exp/Out/) → 1 fichier pour toutes les sorties de tous les processus (mélangés)
 - A venir : allègement des prints dans LMDZ et INCA
 - A venir : 1 fichier concaténant toutes les sorties par processus
- Quelque soit l'état de la simulation (plantée ou finie) les prints de ORCHIDEE sont dans le répertoire \$DMFDIR/IGCM_OUT/.../my_exp/SRF/Debug/ → 1 fichier
- Il faut chercher le message d'erreur indiquant le problème (en faisant une recherche sur les termes « fatal », « error », « severe », « hgardfou » ... en majuscules et minuscules).
Sur titane, ce message est souvent à la fin du fichier

```
forrtl: severe (71): integer divide by zero
Image                PC                Routine                Line                Source                gcm.e
00000000005B924D    Unknown                Unknown                Unknown                Unknown
gcm.e                0000000000576D02    Unknown                Unknown                Unknown
gcm.e                00000000004B7076    Unknown                Unknown                Unknown
gcm.e                00000000004703A5    Unknown                Unknown                Unknown
gcm.e                0000000000450B91    Unknown                Unknown                Unknown
gcm.e                000000000044F67C    Unknown                Unknown                Unknown
libc.so.6            0000003F1021D974    Unknown                Unknown                Unknown
gcm.e                000000000044F549    Unknown                Unknown                Unknown
```

peu d'indications sur la routine dans laquelle est la division par zéro. Pour avoir plus d'informations il faut utiliser l'option « traceback » lors de la compilation

Les niveaux de compilations

- Il y a trois niveaux de compilation pour LMDZ et INCA
 - « -prod » par défaut
 - « -dev »
 - « -debug »

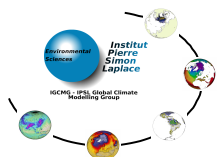
Le niveau de compilation se définit dans le fichier `modips1/config/LMDZORINCA/Makefile`

```
aer_lmdz9695:  
  (cd ../../modeles/INCA3; ./makeinca_fcm ... -debug ... -arch $(FCM_ARCH); .... )  
  (cd ../../modeles/LMDZ4; ./makelmdz_fcm ... -debug ... -arch $(FCM_ARCH) gcm ; ... )
```

Pour connaître plus d'informations sur la routine bugguée il faut activer l'option « -debug »

- Pour les « segmentations fault » vous pouvez envisager les dépassements de tableaux. Pour cela vous devez recompiler le code avec l'option « -check bounds ». Les options de compilations sont dans les fichiers `LMDZ/arch/arch-X64_TITANE.fcm` et `INCA/arch/arch-X64_TITANE.fcm`

<code>%PROD_FFLAGS</code>	<code>-O3</code>	→ option par défaut dans Makefile
<code>%DEV_FFLAGS</code>	<code>-p -g -O3 -traceback -fp-stack-check -ftrapuv</code>	→ option « -dev »
<code>%DEBUG_FFLAGS</code>	<code>-p -g -traceback -check bounds</code>	→ option « -debug »



Le fichier xxx_out_run_file_error

Avec les options « -traceback -check bounds » vous pouvez obtenir un message du type :

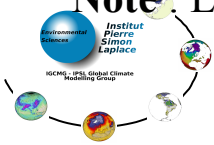
```
forrtl: severe (408): fort: (2): Subscript #1 of the array NEIGHBOURS has value 1 which is greater than the upper bound of 0
```

Image	PC	Routine	Line	Source
gcm.e	000000000180B6ED	Unknown	Unknown	Unknown
gcm.e	000000000180A1F5	Unknown	Unknown	Unknown
gcm.e	00000000017B45D9	Unknown	Unknown	Unknown
gcm.e	000000000176A9EF	Unknown	Unknown	Unknown
gcm.e	000000000176ADF2	Unknown	Unknown	Unknown
gcm.e	000000000F43E96	surf_land_orchide	694	surf_land_orchidee_noopenmp_mod.f90
gcm.e	0000000000F3A852	surf_land_orchide	318	surf_land_orchidee_noopenmp_mod.f90
gcm.e	0000000000F359E9	surf_land_mod_mp_	127	surf_land_mod.f90
gcm.e	0000000000E7D0EE	pbl_surface_mod_m	819	pbl_surface_mod.f90
gcm.e	0000000000667FE7	physiq_	3221	physiq.f
gcm.e	000000000054D5DC	calfis_p_	753	calfis_p.f
gcm.e	00000000004AB316	leapfrog_p_	1021	leapfrog_p.f
gcm.e	0000000000451CCE	MAIN_	759	gcm.f
gcm.e	000000000044F7FC	Unknown	Unknown	Unknown
libc.so.6	0000003F1021D974	Unknown	Unknown	Unknown
gcm.e	000000000044F6C9	Unknown	Unknown	Unknown

Ce qui vous indique que l'on cherche à accéder à l'élément 1 du tableau « neighbours » qui est de taille zéro. Et que le problème se situe à la ligne 694 du fichier **pré-processé** surf_land_orchidee_noopenmp_mod.f90

Note: Les fichiers pré-processés sont dans

- modeles/LMDZ4/libo/.../.config/ppsrc/
- modeles/INCA3/config/ppsrc/



Débugguer un code parallèle

- Vérifier que le problème ne vient pas de la parallélisation en faisant tourner le code sur 1 seul processeur

- Dans config.card : `JobNumProcTot=1`
- Dans Job_xxxx :

```
#PBS -v BATCH_NUM_PROC_TOT=1  
  
...  
if ( [ X${BATCH_NUM_PROC_TOT} != X ] && [ "${BATCH_NUM_PROC_TOT}" -ge 1 ] ) ; then
```

- Dans Makefile :

```
(cd ../../modeles/INCA3; .... -parallel mpi -dev -arch .... ; )  
(cd ../../modeles/LMDZ4; .... -parallel mpi -dev -arch .... ; )
```

- Si les résultats sont les mêmes que sur plusieurs procs → débbugger sur 1 proc
- Si les résultats sont bons alors le problème vient de la parallélisation → débbugger sur plusieurs procs
- Utiliser un débbugueur : ddt → documentation sur le site du ccrt
<https://www-ccrt.ccc.cea.fr/> (login + mot de passe de connection aux machines)



Modifications pour le couplé

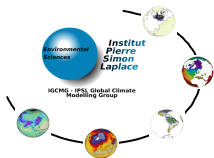
Pour la compilation sur Vargas et Titane

Pour compiler IPSLCM5A sur Vargas ou Titane

- il faut supprimer les 2 clés : "key_vectopt_loop
key_vectopt_memory" dans *config/IPSLCM5A/AA_make*. (à faire avant la commande *ins_make*)
- il faut explicitement demander l'utilisation de 5 processeurs pour NEMO. A faire dans les sources de NEMO.

```
>> vi modipsl/modeles/NEMO/WORK/par_oce.F90 (lignes 29-31)
    jpni   = 1,      &  !: number of processors following i
    jpnj   = 5,      &  !: number of processors following j
    jpnij  = 5          !: nb of local domain = nb of processors
```

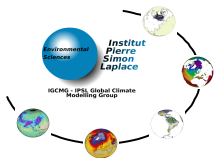
Note : la compilation sur Vargas est très très longue ~ 3h



Modifications pour le couplé pour lancer une simulation sur Vargas

Faire une simulation avec le modèle IPSLCM5A

- Attention : `create_etat0_limit` ne fonctionne pas actuellement sur `vargas`. Il faut donc partir d'un état initial créé autrement : autre simulation, `ce01` exécuté ailleurs, par exemple sur `brodie`.
- Il faut utiliser la commande adéquate de lancement. Dans `config.card` :
`JobRunOptions='"-pgmmodel MPMD -cmdfile"'`
- Il faut préciser dans `PARAM/run.def` : `use_filtre_fft=n`
→ Attention c'est valable pour toute configuration utilisant `LMDZ4_AR5`



<http://forge.ipsl.jussieu.fr/igcmg/wiki/ModipslBeginner1>

Guide d'utilisation des outils de l'ESM

Printable

Commandes de bases pour utiliser un modèle via modipsl :

- en français : [doc_light_french.pdf](#) 
- en anglais : [doc_light_english.pdf](#) 

Nouveautés

- Répertoire Debug/
- Option JobType dans AA_job
- Rebuild sur titane

Environnements de calculs

Accéder à la page

- L'IDRIS
- Le CCRT
- Spécificités propres à :
 - Vargas
 - Brodie
 - Titane
 - Mercure
- Installer son environnement shell **A compléter**
- Le TGCC

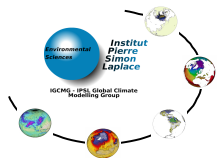
Présentation de modipsl

Accéder à la page

- Extraction de l'architecture modipsl
- Présentation de modipsl
- Liste des configurations de modèles disponibles via modipsl
- Récupérer une configuration donnée

Guide d'utilisation des outils de l'ESM

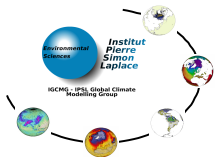
- [Printable](#)
- [Nouveautés](#)
- [Environnements de calculs](#)
- [Présentation de modipsl](#)
- [Compilation](#)
- [Exécution](#)
- [Post-traitement](#)
- [FAQ sur les simulations](#)
- [FAQ sur les post-traitements](#)
- [Debuggage](#)
- [Documentations de modèles forcés et de libIGCM](#)



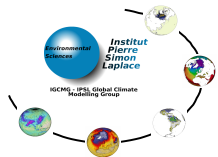
Contributions

L'ensemble du groupe de travail Plate-forme :

esci@ipsl.jussieu.fr



Annexes



Passer de Mercure à Titane

Attention : vos jobs provenant de Mercure ne peuvent pas fonctionner tels quels sur Titane

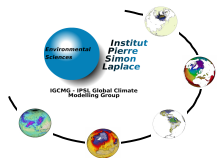
Exemple avec les jobs issus de modipsl

```
#!/bin/ksh
#####
## SX8MERCURE CEA ##
#####
#PBS -N TOTO
#PBS -j o
#PBS -o Script_Output TOTO
#PBS -S /usr/bin/ksh
#PBS -v BATCH_NUM_PROC_TOT=4
#PBS -l cpunum_job=${BATCH_NUM_PROC_TOT}
#PBS -l memsz_job=15.0gb
#PBS -l elapstim_req=01:00:00
```

```
#!/usr/bin/ksh
#####
## TITANE CEA ##
#####
#MSUB -r TOTO
#MSUB -o Script_Output_TOTO
#MSUB -n 32
#MSUB -T 1800 # Limite temps (en secondes)
#MSUB -p gen2211 # groupe de soumission
BATCH_NUM_PROC_TOT=$BRIDGE_MSUB_NPROC
```

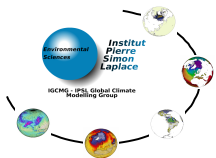
Attention : dans l'entête par défaut le groupe gen2211 est défini. Il faut potentiellement changer le groupe ou le commenter (pour soumettre en tant que dsm)

Note : Pour connaître les groupes auxquels on appartient il faut utiliser la commande « groups »



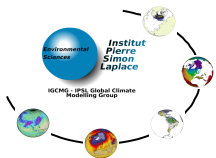
Partir à zéro sur Titane

- Dans ce cas là modipsl fonctionne de la même manière que sur mercure
 - Extraire modipsl
 - Extraire la configuration voulue
 - Créer le makefile
 - Compiler
 - Exécuter
 - Attention au nombre de processus demandés
 - Attention au groupe de soumission (genci, dsm)
 - Attention à la queue de soumission (mono, monoext, test, testext)



Porter un code depuis Mercure vers Titane

- Il est recommandé de récupérer une nouvelle architecture modipsl avec la configuration correspondant à celle que vous voulez porter.
- Ensuite vous effacez les sources des codes (mais pas les scripts) et vous les remplacez par les anciennes
- Vous pouvez ensuite lancer les commandes habituelles (ins_make pour la création de makefile, ins_job pour la création des jobs)



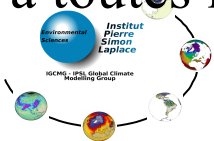
Porter un code depuis Mercure vers Titane

Exemple avec LMDZINCA :

→ On veut porter MERC_MODIPSL sur Titane

```
>> se connecter à Titane (même login et même mot de passe que pour mercure)
>> extraire modipsl
>> cd modipsl/util ; ./model LMDZINCA_v3
>> ./ins_make
>> cd modipsl/modeles/LMDZ4/libf/ ; rm -rf *
>> scp -r ....@mercure.ccc.cea.fr:MERC_MODIPSL/modeles/LMDZ4/libf/* .
>> cd modipsl/modeles/INCA3/src/ ; rm -rf *
>> scp -r ....@mercure.ccc.cea.fr:MERC_MODIPSL/modeles/INCA3/src/* .
>> compiler
```

Vous pouvez également passer le code via le disque cscratch qui est commun à toutes les machines.



Annexe :Caractéristiques d'un fichier NetCDF

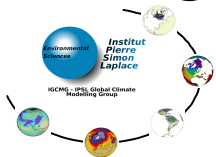
Auto descriptif ➡ Le fichier contient l'information sur les variables contenues

Portable ➡ Fichiers accessibles par des machines ayant des modes différents de stockage des entiers, des caractères et des nombres à virgules flottantes

à Accès direct ➡ Possibilité d'accéder à une donnée sans avoir à parcourir l'ensemble des données qui la précède

Modifiable ➡ Possibilité d'ajouter des données dans un fichier

Partageable ➡ Possibilité d'avoir simultanément un accès en écriture et plusieurs accès en lecture



NetCDF, nco, cdo Convention CF

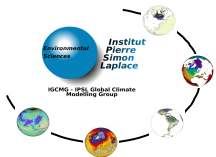
Netcdf : <http://www.unidata.ucar.edu/packages/netcdf>

nco : <http://nco.sourceforge.net/>

cdo : <http://www.mpimet.mpg.de/fileadmin/software/cdo>

Convention CF : <http://www.cgd.ucar.edu/cms/eaton/cf-metadata>

<http://cf-pcmdi.llnl.gov/documents/cf-conventions> (v1.4 stable)



Structure du fichier NetCDF – En-tête

- Informations sur les dimensions



```
dimensions:  
  lon = 72 ;  
  lat = 46 ;  
  presnivs = 19 ;  
  time_counter = UNLIMITED ; // (1 currently)
```

- Informations sur les attributs

(voir conventions CF)



```
// global attributes:  
  :Conventions = "GDT 1.3" ;  
  :file_name = "histmth.nc" ;  
  :production = "An IPSL model" ;  
  :TimeStamp = "2003-MAR-05 10:37:38 GMT+0100" ;  
  :associate_file = "dyn_hist_ave.nc dynzon.nc histhf.nc  
histmth.nc sechiba_out.nc cpl_atm_tauflx.nc cpl_atm_sst.nc" ;
```

ncdump -h
COURS_1m_19790101_197901
30_histmth.nc

- Informations sur les attributs des

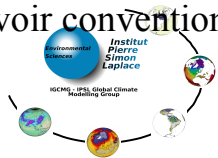
variables

(sans leurs valeurs)

(voir conventions CF)



```
variables:  
  float lon(lon) ;  
    lon:units = "degrees_east" ;  
    lon:valid_min = -180.f ;  
    lon:valid_max = 175.f ;  
    lon:long_name = "Longitude" ;  
    lon:nav_model = "Default grid" ;  
  float lat(lat) ;  
    lat:units = "degrees_north" ;  
    lat:valid_min = -90.f ;  
    lat:valid_max = 90.f ;  
    lat:long_name = "Latitude" ;  
    lat:nav_model = "Default grid" ;  
  float presnivs(presnivs) ;  
    presnivs:units = "mb" ;  
    presnivs:positive = "unknown" ;  
    presnivs:valid_min = 388.2433f ;  
    presnivs:valid_max = 100426.5f ;  
    presnivs:title = "presnivs" ;  
    presnivs:long_name = "Vertical levels" ;  
  float time_counter(time_counter) ;  
    time_counter:units = "seconds since 1979-01-01 00:00:00" ;  
    time_counter:calendar = "360d" ;  
    time_counter:title = "Time" ;  
    time_counter:long_name = "Time axis" ;  
    time_counter:time_origin = " 1979-JAN-01 00:00:00" ;  
  float tsol(time_counter, lat, lon) ;  
    tsol:units = "K" ;  
    tsol:missing_value = 1.e+20f ;  
    tsol:valid_min = 1.e+20f ;  
    tsol:valid_max = -1.e+20f ;  
    tsol:long_name = "Surface Temperature" ;  
    tsol:short_name = "tsol" ;  
    tsol:online_operation = "ave(X)" ;  
    tsol:axis = "TYX" ;  
    tsol:interval_operation = 1800.f ;  
    tsol:interval_write = 2592000.f ;  
    tsol:associate = "time_counter nav_lat nav_lon" ;
```



Structure du fichier NetCDF - Données

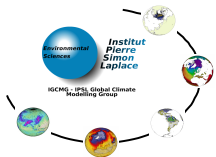
- données de taille fixe
- données de taille variable



data:

```
tsol =  
  246.818, 246.818, 246.818, 246.818, 246.818, 246.818,  
  246.818, 246.818,  
    246.818, 246.818, 246.818, 246.818, 246.818,  
  246.818, 246.818, 246.818,  
    246.818, 246.818, 246.818, 246.818, 246.818,  
  246.818, 246.818, 246.818,  
    246.818, 246.818, 246.818, 246.818, 246.818,  
  246.818, 246.818, 246.818,  
    246.818, 246.818, 246.818, 246.818, 246.818,  
  246.818, 246.818, 246.818,  
    246.818, 246.818, 246.818, 246.818, 246.818,  
  246.818, 246.818, 246.818,  
    246.818, 246.818, 246.818, 246.818, 246.818,  
  246.818, 246.818, 246.818,  
    246.818, 246.818, 246.818, 246.818, 246.818,  
  246.818, 246.818, 246.818,  
    248.3489, 248.3532, 248.3445, 248.003, 247.5628,  
  247.1862, 246.7824,
```

...



Utilitaires nco

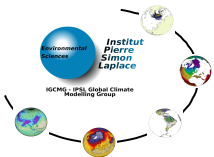
ncdump : génère sur la sortie standard une représentation textuelle CDL d'un ensemble de meta-données netCDF avec la possibilité d'exclure tout ou partie de données variables. La sortie de ncdump doit pouvoir servir d'entrée à ncgen

ncgen : génère un fichier netCDF ou un programme C ou FORTRAN permettant de créer un fichier netCDF

ncdump et ncgen peuvent donc être utilisées comme fonctions inverses pour passer d'une représentation textuelle à une représentation binaire et inversement.

Exemple :

```
> ncdump -p15 -b f  
COURS_1m_19790101_19790130_histmth.nc >  
COURS_1m_19790101_19790130.cdl  
> emacs COURS_1m_19790101_19790130.cdl &  
> ncgen -o COURS_1m_19790101_19790130.nc  
COURS_1m_19790101_19790130.cdl
```

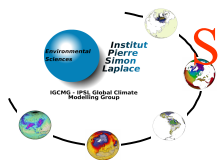


Utilitaires nco

ncdiff soustrait les variables d'un fichier file_1 à celles d'un fichier file_2 correspondantes et stocke les résultats dans un fichier file_3.

ncrcat concatène des variables enregistrées parmi un nombre arbitraire de fichiers d'entrée. La dimension du fichier netCDF de sortie est par défaut la somme des dimensions des fichiers netCDF d'entrée. Les fichiers d'entrée peuvent avoir des tailles différentes mais tous doivent avoir des dimensions spécifiées. L'enregistrement des coordonnées doit avoir la même syntaxe.

Exemple : `ncrcat -v tsol COURS_1m_19790[1-9]01_19790[1-9]30_histmth.nc
COURS_1m_19791[0-2]01_19791[0-2]30_histmth.nc COURS_1m_19880[1-9]01_19880[1-9]30_histmth.nc
COURS_1m_19881[0-2]01_19881[0-2]30_histmth.nc COURS_1m_19790101_19880130_TSOL.nc`



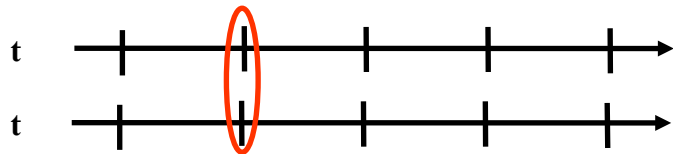
Série temporelle de la variable TSOL sur 10 ans

Utilitaires nco

ngra calcule la moyenne sur un nombre variable de fichiers d'entrée. C'est une moyenne temporelle sur la grille spatiale. Ce qui donne 1 seule valeur dans les fichiers de sorties. **ngra** ne calcule pas de moyenne pondérée.

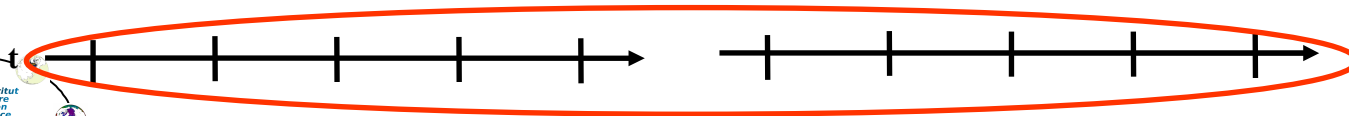
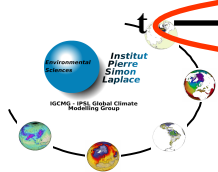
ncea calcule la moyenne sur un nombre variable de fichiers d'entrée. C'est une moyenne spatiale sur la grille temporelle. Ce qui donne autant de valeurs moyennes que de pas de temps.

ncea fait la moyenne « fichier à fichier » sur chaque point de l'axe des temps



Nombre de valeurs de moyenne égale au nombre de sorties par fichiers

ngra fait la moyenne « fichier à fichier » sur l'ensemble des points de l'axe des temps



1 unique valeur de moyenne

Utilitaires nco

ncks permet d'extraire une série de données qu'il écrit sur la sortie standard sous forme ASCII (comme **ncdump**) et qu'il écrit également sous forme d'un fichier binaire netCDF

Exemple :

```
ncks -v sosstsst COURS_1m_19790101_19790130_grid_T.nc  
COURS_1m_19790101_19790130_SOSSTSST.nc
```

...

```
ncks -v sosstsst COURS_1m_19881201_19881230_grid_T.nc  
COURS_1m_19881201_19881230_SOSSTSST.nc
```

```
ncrcat -v sosstsst COURS_1m_19790[1-9]01_19790[1-9]30_grid_T.nc  
COURS_1m_19791[0-2]01_19791[0-2]30_grid_T.nc
```

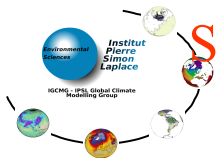
```
    COURS_1m_19880[19]01_19880[1-9]30_grid_T.nc
```

```
    COURS_1m_19881[0-2]01_19881[02]30_grid_T.nc
```

```
COURS_1m_19790101_19880130_grid_T.nc
```

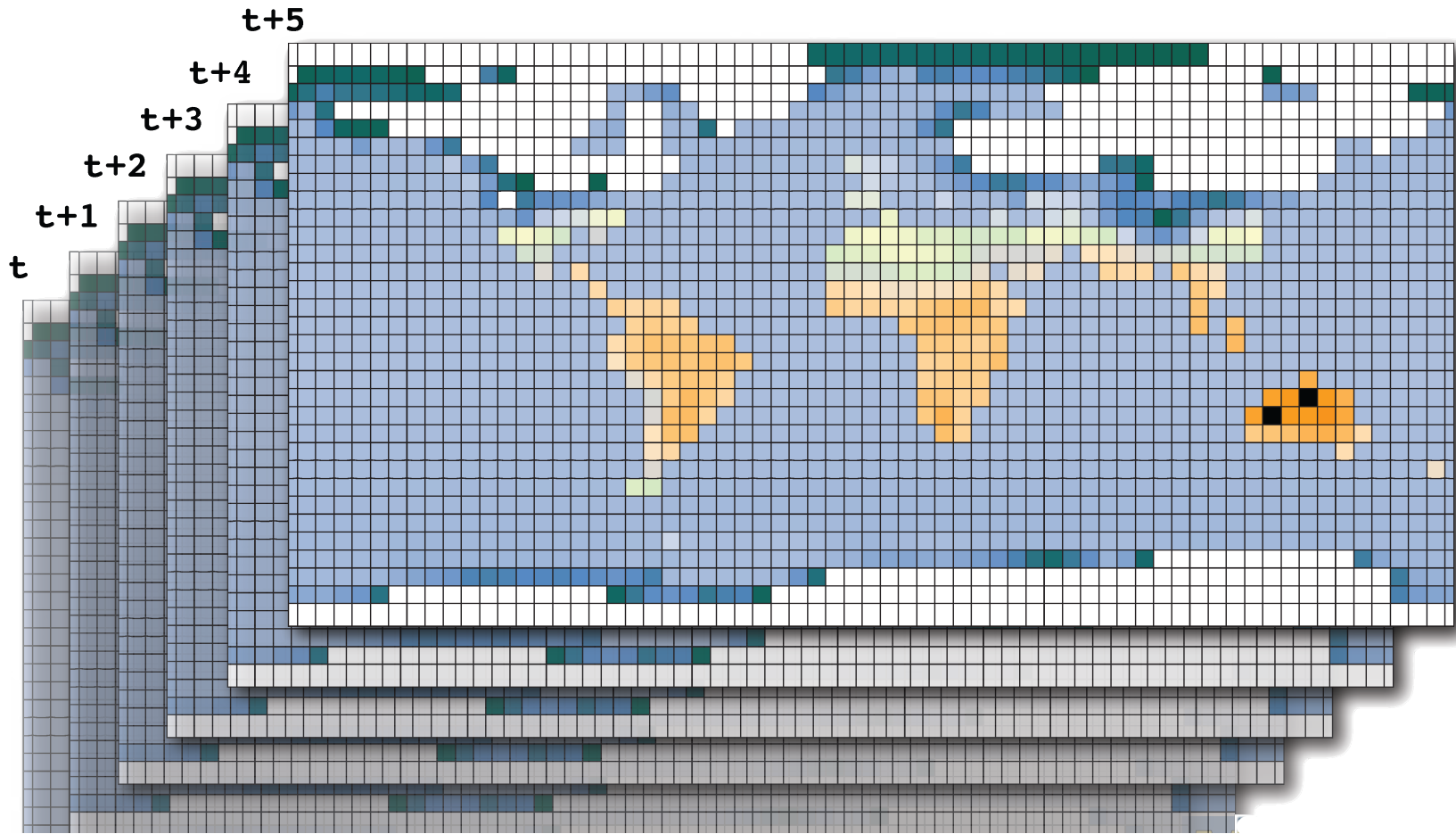


Série temporelle de la variable SOSSTSST sur 10 ans



Utilitaires cdo (Climate Data Operator)

2d fields timeseries

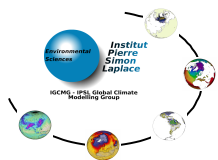
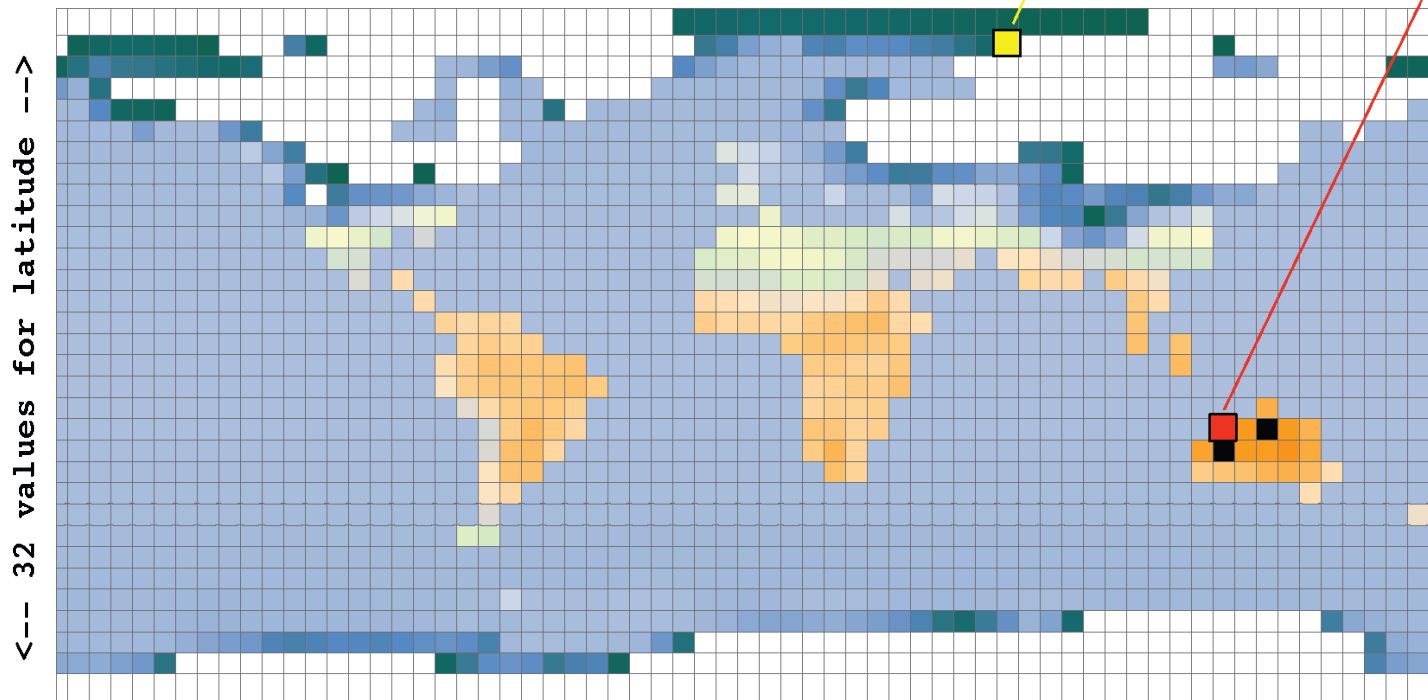


Utilitaires cdo (Climate Data Operator)

```
tiny 23%cdo info tsurf.1.nc
```

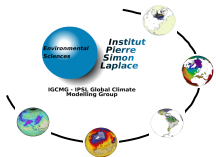
```
-1 :      Date Time Code Level Size Miss : Minimum Mean Maximum  
1 : 1978-01-02 00:00 169      0 2048  0 : 226.98 268.46 311.08  
cdo info : Processed 1 variable 1 timestep. ( 0.00s )
```

<-- 64 values for longitude -->



Utilitaires cdo (Climate Data Operator)

- File information (info, sinfo, diff, diffv, ...)
- File operations (copy, cat, merge, split, ...)
- Selection (selcode, selvar, sellevel, seltimestep, ...)
- Missing values (setctomiss, setmisstoc, setrtomiss)
- Arithmetic (add, sub, mul, div, ...)
- Mathematical functions (sqrt, exp, log, sin, cos, ...)
- Comparison (eq, ne, le, lt, ge, gt, ...)
- Conditions (ifthen, ifnotthen, ifthenc, ifnotthenc)
- Field statistic (fldsum, fldavg, fldstd, fldmin, fldmax, ...)
- Vertical statistic (vertsum, vertavg, vertstd, vertmin, ...)
- Time range statistic (timavg, yearavg, monavg, dayavg, ...)
- Ensemble statistic (enssum, ensavg, ensstd, ensmin, ...)
- Regression (detrend)
- Field interpolation (remapbil, remapcon, remapdis, ...)
- Vertical interpolation (ml2pl, ml2hl)
- Time interpolation (inttime, intyear)



Job_ *JobName* : PBS tableau des classes* IDRIS

brodie : news class

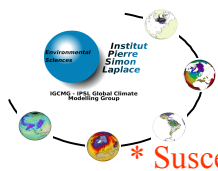
=====
Classes multiprocesseurs (<=8) au sein d'un noeud (MPI ou OpenMP)
=====

Parametres NQSII a specifier :

#PBS -q multi

#PBS -l cpunum_job=<Nproc> # Nombre de processeurs (1 <= Nproc <= 8)

```
      ^ -l cputim_job (limite en temps CPU par job)
      |
12:00:00 +-----+
      (12H) |
            |      p2t2
            |
            | 1 <= Nproc <= 2
            | TMPDIR <= 45Gb
            |
1:00:00 +-----+
      (1H) |
            |      p2t1
            |
            | 1 <= Nproc <= 2
            | TMPDIR <= 45Gb
            |
      +-----+----> -l memsz_job (limite memoire par job)
                        15Gb
```

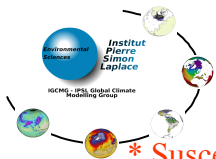


* Susceptible de changement permanent

Job *JobName* : PBS tableau des classes* IDRIS

brodie : news class (suite)

```
^ -l cputim_job
|
48:00:00 +-----+
(48H) |
|           p8t2
|
|           7 <= Nproc <= 8
|           TMPDIR <= 300Gb
|
2:00:00 +-----+
(2H) |
|           p8t1
|
|           7 <= Nproc <= 8
|           TMPDIR <= 100Gb
|
+-----> -l memsz_job
|
|           60Gb
```



* Susceptible de changement permanent

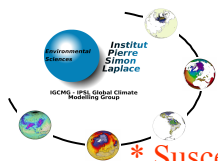
Job_JobName : PBS

tableau des classes* CCRT

mercure : class

Classe de soumission:

Queue	Act	Ena	Nod	Cpu	Mem	Time	Type	RL	URL	UAL	Hosts
scal-new	ENA	ACT	1	1	8G	24h	-	-	-	-	mercure02,mercure03
scalaire	ENA	ACT	1	1	8G	24h	-	-	-	-	mercure
admin-sx	ENA	ACT	8	8	52G	Inf	-	-	-	-	mercure10,mercure11,... mercure17
admin-sx	ENA	ACT	3	16	962G	Inf	-	-	-	-	mercure20,mercure21,mercure22
para- sx9	ENA	ACT	3	16	962G	24h	Normal	-	-	-	mercure20,mercure21,mercure22
prio- sx9	ENA	ACT	3	16	962G	24h	Special	-	-	-	mercure20,mercure21,mercure22
test- sx9	ENA	ACT	3	16	962G	1h	Urgent	-	-	1	mercure20,mercure21,mercure22
bigmem	ENA	ACT	1	1	64G	24h	Normal	-	-	2	mercure10,mercure11
bigtime	ENA	ACT	1	1	32G	100h	Normal	-	-	2	mercure12,mercure13,... mercure16
parallel	ENA	ACT	4	8	40G	24h	Normal	-	-	2	mercure10,mercure11,... mercure16
prod	ENA	ACT	1	1	20G	24h	Normal	-	-	12	mercure10,mercure11,... mercure17
test	ENA	ACT	1	1	32G	1h	Urgent	8	1	-	mercure17
testpara	ENA	ACT	4	8	52G	0.5h	Urgent	1	1	-	mercure10,mercure11,... mercure16



* Susceptible de changement permanent